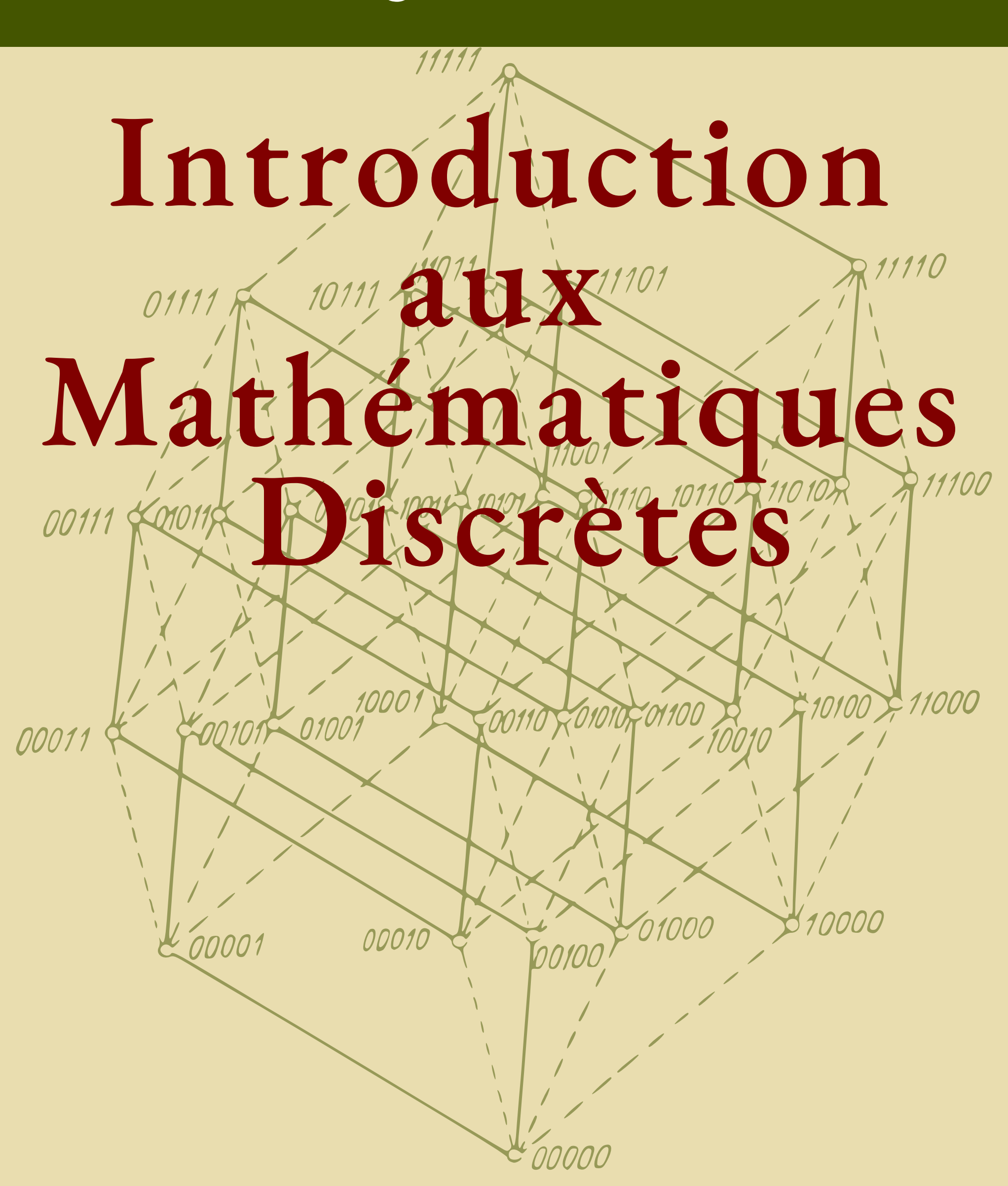
Serge Yablonski



Introduction aux Mathématiques Discrètes

Éditions Mir Moscou

S. YABLONSKI

INTRODUCTION AUX MATHÉMATIQUES DISCRÈTES

ÉDITIONS MIR · MOSCOU

Traduit du russe
par DJILALI EMBAREK

на французском языке

© Главная редакция физико-математической литературы
издательства « Наука » 1979

© Traduction française Editions Mir 1983

TABLE DES MATIÈRES

AVANT-PROPOS

Les mathématiques discrètes plongent leurs racines dans la plus haute antiquité. Comme leur nom l'indique, elles sont l'antithèse des mathématiques « continues ». Dans un sens large les mathématiques discrètes comprennent des disciplines « mûres » telles la théorie des nombres, l'algèbre, la logique mathématique ainsi que des disciplines qui ont commencé à se développer le plus intensément au milieu de ce siècle par suite du progrès scientifique et technique qui a mis à l'ordre du jour l'étude de systèmes commandés à l'aide des calculateurs. Dans un sens étroit les mathématiques discrètes restreignent leur champ à l'étude de ces nouvelles disciplines. C'est précisément dans cet esprit que nous les comprenons en rapportant à ces nouvelles disciplines la théorie des systèmes fonctionnels, la théorie des graphes et des réseaux, la théorie du codage, l'analyse combinatoire, la programmation en nombres entiers, etc.

Les mathématiques discrètes ne constituent pas seulement une base de la cybernétique, elles sont aussi un important maillon dans la culture mathématique. Le contenu de cet ouvrage correspond à deux types de programmes du cours « Mathématiques discrètes »: un programme standard pour les facultés de mathématiques appliquées et de cybernétique et un programme calqué sur le cours lu à l'Université de Moscou. Ces programmes ne se fixent pas pour objectif de développer des problèmes d'actualités. Une spécialisation excessive et un assujettissement des programmes à des faits spéciaux présentent un risque, celui d'une dévalorisation de ces faits au bout de 10 à 15 ans (ce qui coïncide précisément avec la plus intense période d'activités des étudiants qui ont suivi ce cours). Pour cette raison, le but majeur de cet ouvrage est l'initiation aux méthodes de raisonnement propres aux mathématiques discrètes. Cet ouvrage familiarise le lecteur avec les problèmes clefs de quelques domaines des mathématiques discrètes et de leurs applications. L'exposé est conçu de manière à réduire au minimum le nombre des notions nécessaires et, d'autre part, à donner quelques dix à quinze théorèmes importants avec des démonstrations différentes, à faire

enfin connaître les applications de la notion d'algorithme dont la maîtrise est primordiale pour les mathématiciens appliqués.

Ce livre a son origine dans le cours lu par l'auteur à la faculté de mécanique et de mathématiques de l'Université de Moscou en 1964.

Cet ouvrage comprend quatre parties.

Première partie. Systèmes de fonctions munis d'opérations.

Deuxième partie. Graphes et réseaux.

Troisième partie. Théorie du codage.

Quatrième partie. Quelques applications à la cybernétique.

De nombreuses questions ne sont pas traitées sous une forme abstraite: on fait largement appel au langage géométrique et aux interprétations suggestives. Ceci permet de combiner dans les constructions la rigueur à la suggestion. Il faut dire que deux points de vue extrêmes prévalent en mathématiques: le manque de rigueur et la rigueur à outrance. Ces tendances sont aussi dangereuses l'une que l'autre. La rigueur doit correspondre au problème étudié (et au niveau de l'auditoire) au même titre que dans les calculs approchés le nombre de décimales doit correspondre à la précision des données initiales. Un « excès » de rigueur est dans cet ordre d'idées comparable aux calculs comprenant un nombre élevé de décimales.

Les démonstrations de nombreux théorèmes sont dues à l'auteur et sont publiées pour la première fois.

Cet ouvrage est destiné aux étudiants, élèves du second et du troisième cycle ainsi qu'aux chercheurs.

S. Yablonski

PREMIÈRE PARTIE

SYSTÈMES DE FONCTIONS MUNIS D'OPÉRATIONS

Les systèmes de fonctions munis d'opérations constituent l'une des branches des mathématiques discrètes, qui se développe le plus intensément. Le principal objet de notre étude sera les fonctions discrètes: fonctions booléennes, fonctions de la logique k-valente, fonctions calculables, etc. Cette partie présente des affinités avec l'analyse mathématique dans la mesure où elle traite de problèmes similaires: propriétés des fonctions, expression d'une fonction au moyen d'un ensemble (généralement fini) de « fonctions élémentaires », etc.

CHAPITRE PREMIER

ALGÈBRE LOGIQUE

§ 1. Fonctions de l'algèbre logique

Soit $U = \{u_1, u_2, \ldots, u_m, \ldots\}$ un alphabet de variables (d'arguments). On se propose d'étudier des fonctions $f(u_{i_1}, u_{i_2}, \ldots, u_{i_n})$ ($u_{i_\nu} \neq u_{i_\mu}$ pour $\nu \neq \mu$) dont les arguments sont définis sur l'ensemble $E^2 = \{0, 1\}$ et telles que $f(\alpha_1, \alpha_2, \ldots, \alpha_n) \in E^2$, lorsque $\alpha_i \in E^2$ ($i = 1, 2, \ldots, n$). Ces fonctions seront appelées *fonctions de l'algèbre logique* ou *fonctions booléennes.* Pour alléger les notations des indices des variables, on prendra pour métasymboles les symboles $x, y, z, \ldots$ de même que ces symboles avec des indices. Ainsi, $f(x_1, x_2, \ldots, x_n)$ doit être comprise comme une fonction dépendant d'arguments arbitraires $u_{i_1}, u_{i_2}, \ldots, u_{i_n}$, où $u_{i_\nu} \neq u_{i_\mu}$ pour $\nu \neq \mu$.

De la définition de la fonction $f(x_1, x_2, \ldots, x_n)$ il s'ensuit que pour la donner il suffit d'en indiquer les valeurs qui correspondent à chaque combinaison de valeurs des arguments, c'est-à-dire de dresser un tableau (voir tableau 1). Il est immédiat de voir que n variables prennent 2^n valeurs distinctes. Pour la commodité on utilisera la disposition standard des combinaisons: c'est-à-dire qu'on considère qu'une combinaison est la représentation binaire d'un nombre et

l'on ordonne ces combinaisons comme les nombres 0, 1, . . ., $2^n - 1$. On voit d'autre part que chaque fonction $f(x_1, x_2, \ldots, x_n)$ définit une application

$$\underbrace{E^2 \times E^2 \times \ \ldots \ \times E^2}_{n \text{ fois}} \to E^2.$$

Il est donc naturel d'interpréter le symbole f comme le symbole de cette application et $x_1, x_2, \ldots, x_n$, comme les noms des colonnes. Ceci étant, les fonctions

$$f(x_1, x_2, \ldots, x_n), \quad f(y_1, y_2, \ldots, y_n)$$

définiront la même application et leurs tableaux ne se distingueront éventuellement que par les noms des colonnes. Désignons par P_2

Tableau 1

$x_1 \ldots x_{n-1}, x_n$	$f(x_1, \ldots, x_{n-1}, x_n)$
0 ... 0 0	f(0, ..., 0, 0)
0 ... 0 1	f(0, ..., 0, 1)
0 ... 1 0	f(0, ..., 1, 0)
. . . .	. . . .
1 ... 1 1	f(1, ..., 1, 1)

le système de toutes les fonctions booléennes sur l'alphabet U, qui contient aussi les constantes 0 et 1.

Si l'on fige n variables $x_1, x_2, \ldots, x_n$, les tableaux ne se distingueront que par les valeurs de la colonne de droite. D'où le

Théorème 1. *Le nombre $p_2(n)$ de toutes les fonctions de P_2 des variables $x_1, x_2, \ldots, x_n$ est égal à 2^{2^n}.*

Ce théorème appelle deux remarques.

1. Les fonctions booléennes de n variables données sont en nombre fini. Donc, pour savoir si les fonctions de cet ensemble fini jouissent de telle ou telle propriété, il suffit d'examiner chaque fonction de cet ensemble. Or, les nombres $p_2(n)$ croissent très vite avec n:

$$p_2(1) = 4, \quad p_2(2) = 16, \quad p_2(3) = 256, \quad p_2(4) = 65536, \ \ldots$$

Donc, pour des valeurs relativement peu élevées de n ($n \geqslant 6$) cet examen devient pratiquement impossible même avec le concours des calculateurs.

2. Le tableau définissant la fonction se complique singulièrement lorsque le nombre des arguments croît. Ainsi, par exemple, pour un nombre d'arguments peu élevé, disons pour $n = 10$, le tableau compte

1024 lignes ; pour $n = 20$ la taille du tableau prend des proportions démesurées.

La notion de fonction introduite plus haut est incomplète en ce sens qu'elle ne permet pas de traiter des fonctions de n variables comme des fonctions de plus de n variables. Pour obvier à ce défaut, introduisons la définition suivante.

Définition. On dit qu'une fonction $f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots \ldots, x_n)$ de P_2 *dépend essentiellement* d'un argument x_i s'il existe des valeurs $\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n$ des variables $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$, telles que

$$f(\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n) \neq \neq f(\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n).$$

La variable x_i est dite *essentielle*. Si x_i n'est pas essentielle, on l'appelle *inessentielle* ou *fictive*.

Supposons que la variable x_i est inessentielle pour la fonction $f(x_1, \ldots, x_n)$. Considérons le tableau de la fonction $f(x_1, \ldots, x_n)$ et construisons un nouveau tableau en biffant toutes les lignes de la forme $(\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n)$ et la colonne de l'argument x_i. Le tableau obtenu définira une fonction $g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots \ldots, x_n)$. On dira que la fonction $g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ a été déduite à partir de $f(x_1, \ldots, x_n)$ *par élimination de la variable inessentielle* x_i ou inversement que la fonction $f(x_1, \ldots, x_n)$ s'obtient à partir de $g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ *par adjonction de la variable inessentielle* x_i.

Définition. Deux fonctions $f_1(x_1, \ldots, x_n)$ et $f_2(x_1, \ldots, x_n)$ sont dites *égales* si elles se déduisent l'une de l'autre par adjonction ou élimination de variables inessentielles.

Dans la suite, toutes les fonctions seront considérées à des variables inessentielles près, c'est-à-dire que la donnée d'une fonction f_1 équivaudra à la donnée d'une fonction quelconque f_2 égale à f_1. L'égalité des fonctions sera notée par le symbole fonctionnel $\equiv$. Ainsi $f_1 \equiv f_2$.

Il existe deux types de fonctions ne possédant pas de variables essentielles : les fonctions identiquement égales à 0 et les fonctions identiquement égales à 1. De ce fait il semble logique d'étudier les constantes 0 et 1 (en les traitant comme des fonctions d'un ensemble vide de variables).

Remarque. Si est donné un système fini de fonctions de P_2 : $\{f_1, \ldots, f_s\}$, $s \geqslant 1$, on peut considérer que ces fonctions dépendent toutes des mêmes variables $x_1, \ldots, x_n$, c'est-à-dire qu'elles sont de la forme $f_1(x_1, \ldots, x_n), \ldots, f_s(x_1, \ldots, x_n)$.

Voyons en conclusion de ce paragraphe des exemples de fonctions booléennes. Ces fonctions sont d'un usage courant en logique mathématique et en cybernétique et jouent le même rôle que, par exemple, les fonctions x^n ou sin x en analyse, c'est pourquoi on peut les qualifier de fonctions « élémentaires »:

1) $f_1(x) = 0$: la *constante* 0;

2) $f_2(x) = 1$: la *constante* 1;

3) $f_3(x) = x$: la *fonction identique*;

4) $f_4(x) = \bar{x}$: la *négation de* x ($\bar{x}$ se lit « non x »);

5) $f_5(x_1, x_2) = (x_1 \,\&\, x_2)$: la *conjonction de* x_1 *et de* x_2 (on lit « x_1 et x_2 »). Au lieu de & on écrit · ou bien on omet tout simplement ce signe et on écrit (x_1x_2). Cette fonction est souvent appelée *produit logique*;

6) $f_6(x_1, x_2) = (x_1 \vee x_2)$: la *disjonction de* x_1 *et* x_2 (lire « x_1 ou x_2 »). Cette fonction est souvent appelée *somme logique*;

Tableau 2

x	0	1	x	$\bar{x}$
0	0	1	0	1
1	0	1	1	0

7) $f_7(x_1, x_2) = (x_1 \to x_2)$: l'*implication de* x_2 *par* x_1 (lire « x_1 entraîne x_2 »);

8) $f_8(x_1, x_2) = (x_1 + x_2)$: l'*addition* mod. 2 *de* x_1 *et de* x_2;

9) $f_9(x_1, x_2) = (x_1 \mid x_2)$: la *fonction de Sheffer* (« incompatibilité »).

Tableau 3

x_1	x_2	$(x_1 \,\&\, x_2)$	$(x_1 \vee x_2)$	$(x_1 \to x_2)$	$(x_1 + x_2)$	$(x_1 \mid x_2)$
0	0	0	0	1	0	1
0	1	0	1	1	1	1
1	0	0	1	0	1	1
1	1	1	1	1	0	0

Les tableaux 2 et 3 représentent les valeurs de ces fonctions. On remarquera que

$$(x_1 \,\&\, x_2) = \min(x_1, x_2) = (x_1 \cdot x_2), \quad (x_1 \vee x_2) = \max(x_1, x_2).$$

§ 2. Formules. Réalisation des fonctions par des formules

Comme en algèbre élémentaire on peut construire des formules à partir de fonctions élémentaires. Voici une définition des formules par récurrence.

Définition. Soit $\mathfrak{P}$ un sous-ensemble (pas forcément fini) de fonctions de P_2.

a) **Base de la récurrence.** Chaque fonction $f(x_1, \ldots, x_m)$ de $\mathfrak{P}$ s'appelle *formule sur* $\mathfrak{P}$.

b) **Passage de la récurrence.** Soient $f(x_1, \ldots, x_m)$ une fonction de $\mathfrak{P}$, $A_1, \ldots, A_m$ des expressions qui sont soit des formules sur $\mathfrak{P}$, soit des symboles de variables de U. Alors l'expression $f(A_1, \ldots, A_m)$ s'appelle *formule sur* $\mathfrak{P}$.

Exemple 1. Soit $\mathfrak{P}$ un ensemble de fonctions « élémentaires ». Les expressions suivantes sont des formules sur $\mathfrak{P}$:

1) $\{[(x_1x_2) + x_1] + x_2\}$;

2) $[\bar{x}_1(x_2 + x_3)]$;

3) $\overline{\{x_1 \vee [(x_2 \to x_3)(x_3 \to x_2)]\}}$.

Dans la suite on désignera les formules par les lettres gothiques majuscules avec des crochets contenant les fonctions nécessaires à leur construction. Ainsi

$$\mathfrak{A}[f_1, \ldots, f_s]$$

signifie que la formule $\mathfrak{A}$ a été construite avec les fonctions $f_1, \ldots, f_s$. Si l'on veut mentionner l'ensemble des variables qui ont participé à la construction de la formule, on écrit

$$\mathfrak{A}(x_1, \ldots, x_n).$$

$\mathfrak{A}$ étant une formule sur $\mathfrak{P}$, on appellera *sous-formules* de $\mathfrak{A}$ les formules qui ont servi à construire $\mathfrak{A}$.

Soit $\mathfrak{A}$ une formule sur un ensemble $\mathfrak{P} = \{f_1(x_1, \ldots, x_n), \ldots, f_s(x_1, \ldots, x_n)\}$, c'est-à-dire que $\mathfrak{A} = \mathfrak{A}[f_1, \ldots, f_s]$. Considérons l'ensemble de fonctions

$$\mathfrak{Q} = \{g_1(x_1, \ldots, x_n), \ldots, g_s(x_1, \ldots, x_n)\},$$

où g_i dépend des mêmes variables que f_i $(i = 1, \ldots, s)$.

Définition. Considérons la formule $\mathfrak{B} = \mathfrak{B}[g_1, \ldots, g_s]$ obtenue à partir de $\mathfrak{A}$ par le changement $\begin{pmatrix} f_1, & \ldots, & f_s \\ g_1, & \ldots, & g_s \end{pmatrix}$. On dit que la formule $\mathfrak{B}$ est de *même structure* que la formule $\mathfrak{A}$.

Exemple 2.

1) $\mathfrak{P} = \{\bar{x}_1, (x_1 \& x_2)\}$, $\mathfrak{A} = [x_1 \& \overline{(x_2 \& x_3)}]$;

2) $\mathfrak{Q} = \{\bar{x}_1, (x_1 \to x_2)\}$, $\mathfrak{B} = [x_1 \to \overline{(x_2 \to x_3)}]$.

Il est évident que $\mathfrak{A}$ et $\mathfrak{B}$ ont la même structure.

Dans la suite on désignera la structure d'une formule par S (avec ou sans indices). La formule $\mathfrak{A}$ est définie de façon unique par la structure S et l'ensemble ordonné $\{f_1, f_2, \ldots, f_s\}$. On peut donc écrire $\mathfrak{A} = S[f_1, f_2, \ldots, f_s]$.

Associons maintenant à toute formule $\mathfrak{A}(x_1, \ldots, x_n)$ sur $\mathfrak{P}$ une fonction $f(x_1, \ldots, x_n)$ de P_2 en utilisant la définition des formules par récurrence.

a) **Base de la récurrence.** Si $\mathfrak{A}(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$, où $f \in \mathfrak{P}$, associons la fonction $f(x_1, \ldots, x_n)$ à la formule $\mathfrak{A}(x_1, \ldots, x_n)$.

b) **Passage de la récurrence.** Supposons que $\mathfrak{A}(x_1, \ldots, x_n) = f_0(A_1, \ldots, A_m)$, où A_i $(i = 1, \ldots, m)$ est soit une formule sur $\mathfrak{P}$, soit le symbole d'une variable $x_{j(i)}$. Alors, d'après l'hypothèse de la récurrence, à A_i correspond soit une fonction f_i de P_2, soit la fonction identique $f_i = x_{j(i)}$. Associons à la formule $\mathfrak{A}(x_1, \ldots, x_n)$ la fonction $f(x_1, \ldots, x_n) = f_0(f_1, \ldots, f_m)$.

Si une fonction f est associée à une formule $\mathfrak{A}$, on dit aussi que la formule $\mathfrak{A}$ *réalise* ou *engendre* ou *représente* f.

La fonction f correspondant à la formule $\mathfrak{A}$ sera appelée *superposition* de fonctions de $\mathfrak{P}$, et la procédure donnant la fonction f de $\mathfrak{P}$, *opération de superposition.*

Exemple 3. Soient $f_1(x_1, x_2)$, $f_2(x_1, x_2, x_3)$ et $f_3(x_1, x_2, x_3)$ des fonctions réalisées par les formules de l'exemple 1.

1) La formule $(((x_1x_2) + x_1) + x_2)$ se construit en trois pas. Nous avons les sous-formules suivantes:

$$(x_1x_2),\ ((x_1x_2) + x_1),\ (((x_1x_2) + x_1) + x_2).$$

Le tableau 4 représente les fonctions réalisées par ces formules. Ces fonctions ont été calculées à l'aide du tableau 3 et de la règle b). La dernière colonne définit la fonction $f_1(x_1, x_2)$; il est évident que $f_1(x_1, x_2) = \max(x_1, x_2)$.

Tableau 4

x_1	x_2	(x_1x_2)	$((x_1x_2) + x_1)$	$(((x_1x_2) + x_1) + x_2)$
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	0	1

2) Nous allons construire la fonction $f_2(x_1, x_2, x_3)$ d'une autre manière (qui résulte aussi de la définition). Pour toute combinaison $(\sigma_1, \sigma_2, \sigma_3)$, en utilisant les tableaux 2 et 3, on trouve

$$f_2(\sigma_1, \sigma_2, \sigma_3) = (\bar{\sigma}_1(\sigma_2 + \sigma_3))$$

(voir tableau 5).

Tableau 5

x_1	x_2	x_3	$f_2(x_1, x_2, x_3)$	$f_3(x_1, x_2, x_3)$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

3) Pour déterminer la fonction $f_3(x_1, x_2, x_3)$, on se servira des tableaux 2 et 3 pour trouver les combinaisons sur lesquelles la formule

$$\overline{\{x_1 \vee [(x_2 \to x_3)(x_3 \to x_2)]\}}$$

prend la valeur 1. Il est évident que cela revient à trouver les combinaisons des variables pour lesquelles la formule

$$\{x_1 \vee [(x_2 \to x_3)(x_3 \to x_2)]\}$$

est égale à 0. Ceci aura lieu pour $x_1 = 0$ et dans les cas où $[(x_2 \to x_3)(x_3 \to x_2)]$ prend la valeur 0. Enfin, la formule $[(x_2 \to x_3)(x_3 \to x_2)]$ est égale à 0 si l'une au moins des formules $(x_2 \to x_3)$, $(x_3 \to x_2)$ l'est, ce qui a lieu pour $x_2 = 1$, $x_3 = 0$ ou pour $x_2 = 0$, $x_3 = 1$. Donc, $f_3(x_1, x_2, x_3) = 1$ sur les combinaisons (001) et (010) (cf. tableau 5).

Remarque 1. Dans la définition des formules, le numéro b) peut être remplacé par un numéro utilisant deux opérations plus simples.

1) *Substitution des variables.* Soit

$$S = \begin{pmatrix} x_1, & \ldots, & x_n \\ x_{i_1}, & \ldots, & x_{i_n} \end{pmatrix}$$

une substitution des variables (x_{i_ν} n'est pas forcément différent de x_{i_μ} pour $\nu \neq \mu$). Cette substitution permet de changer les variables de la fonction $f(x_1, \ldots, x_n)$ et d'obtenir en définitive la fonction

$f(x_{i_1}, \ldots, x_{i_n})$. Il est évident que la substitution consiste à débaptiser, permuter et identifier les variables.

2) *Substitution non itérative de fonctions.* Elle permet de former des expressions $f(A_1, \ldots, A_m)$, où A_i est soit une formule, soit une variable de U, de plus l'une au moins des A_i n'est pas une variable et les ensembles des variables figurant dans A_i et A_j ne se rencontrent pas.

Il est évident que toute formule sur $\mathfrak{P}$ peut être obtenue à partir de fonctions appartenant à $\mathfrak{P}$ par une substitution non itérative de fonctions suivie d'une substitution des variables.

Remarque 2. Si $\mathfrak{P}$ contient la fonction identique, le numéro b) de la définition des formules et des fonctions réalisables par des formules s'énonce sous une forme plus simple : il faut supposer que les A_i sont toutes des formules sur $\mathfrak{P}$.

Le langage des formules introduit est commode pour l'écriture des fonctions booléennes décrivant diverses conditions ou propositions. Illustrons ceci sur deux exemples. Nous allons utiliser des propositions de la forme « x a lieu » ou tout simplement « x », ce qui à son tour signifie que pour les conditions données, x est vrai ou égal à 1.

Exemple 4. *Addition des nombres binaires à n chiffres.* Nous partons de l'algorithme ordinaire d'addition rang par rang

$$\begin{array}{r} x_n \ldots x_2x_1 \\ + \quad y_n \ldots y_2y_1 \\ \hline z_{n+1}z_n \ldots z_2z_1 \end{array}\ .$$

On demande d'exprimer les valeurs des chiffres de la somme en fonction de ceux des termes. Pour résoudre ce problème on considère des variables auxiliaires $w_n, w_{n-1}, \ldots, w_1$, où w_i désigne le report du rang i au rang $i+1$. Ces paramètres apparaissent dans l'algorithme mentionné.

Il est clair alors que

$$z_i = ((x_i + y_i) + w_{i-1})$$
$$(w_0 = 0, \quad x_{n+1} = y_{n+1} = 0, \quad i = 1, \ldots, n+1).$$

La variable w_i est déterminée par la condition de report du rang i au rang $i+1$: « le report au rang $i+1$ a lieu si et seulement si deux au moins des trois variables x_i, y_i, w_{i-1} sont égales à 1 ». Cette proposition s'énonce de la manière plus détaillée suivante : « x_i et y_i » ou « x_i et w_{i-1} » ou « y_i et w_{i-1} ».

Si maintenant on remplace les symboles « et » et « ou » par & et $\vee$ on obtient la formule suivante pour w_i :

$$w_i = \{[(x_i \,\&\, y_i) \vee (x_i \,\&\, w_{i-1})] \vee (y_i \,\&\, w_{i-1})\},$$
$$(i = 1, \ldots, n).$$

Exemple 5. *Problème de l'appel d'un ascenseur libre.* Supposons qu'un immeuble de n étages soit desservi par trois ascenseurs. A chaque étage il existe un dispositif permettant d'appeler le plus proche ascenseur libre. On demande d'écrire en langage logique les conditions d'appel de l'ascenseur i ($i = 1, 2, 3$). Nous étudions ce problème pour le cas où l'appel a lieu au premier étage.

L'information initiale se décrit à l'aide des $3n$ variables

$$x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_n, y_n, z_n,$$

où $x_i = 1$ si et seulement si l'ascenseur 1 se trouve à l'étage i et est libre ; $y_i = 1$ si et seulement si l'ascenseur 2 se trouve à l'étage i et est libre ; $z_i = 1$ si et seulement si l'ascenseur 3 se trouve à l'étage i et est libre.

Désignons par

$$f_{\mathrm{I}}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n), \ldots, f_{\mathrm{III}}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n)$$

des fonctions égales à 1 si et seulement si est appelé l'ascenseur respectivement de numéro 1, 2, 3. La condition d'appel de l'ascenseur 1, ou fonction f_{I}, s'énonce : « l'ascenseur 1 est libre et il n'en existe pas d'autres à un étage inférieur ». De façon plus détaillée : « l'ascenseur 1 est appelé si et seulement s'il se trouve au premier étage et est libre ou au premier étage les ascenseurs 2 et 3 ne sont pas libres, auquel cas l'ascenseur 1 se trouve au deuxième étage et est libre ou au deuxième étage les ascenseurs 2 et 3 ne sont pas libres et alors . . . ». Exprimons cette proposition en fonction de $x_1, y_1, z_1, \ldots, x_n, y_n, z_n$: « l'ascenseur 1 est appelé si et seulement si x_1 ou si « non y_1 et non z_1 », auquel cas x_2 ou « non y_2 et non z_2 » et alors . . . ». On obtient aisément la formule f_{I} en remplaçant maintenant les conjonctions « et » et « ou » par & et $\vee$ et en disposant les parenthèses conformément aux liaisons :

$$f_{\mathrm{I}}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n) = \\ = \{x_1 \vee \{(\bar{y}_1 \,\&\, \bar{z}_1) \,\&\, [x_2 \vee [(\bar{y}_2 \,\&\, \bar{z}_2) \,\&\, (\ldots)]]\}\}.$$

De façon analogue, on obtient pour les fonctions f_{II} et f_{III} :

$$f_{\mathrm{II}}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n) = \\ = \{y_1 \vee \{(\bar{x}_1 \,\&\, \bar{z}_1) \,\&\, [y_2 \vee [(\bar{x}_2 \,\&\, \bar{z}_2) \,\&\, (\ldots)]]\}\},$$

$$f_{\mathrm{III}}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n) = \\ = \{z_1 \vee \{(\bar{x}_1 \,\&\, \bar{y}_1) \,\&\, [z_2 \vee [(\bar{x}_2 \,\&\, \bar{y}_2) \,\&\, (\ldots)]]\}\}.$$

On remarque donc que le langage des formules présente un intérêt indéniable.

2—0382

§ 3. Equivalence des formules. Propriétés des fonctions élémentaires. Principe de dualité

Nous avons vu que toute formule sur $\mathfrak{P}$ engendre une fonction booléenne, mais que différentes formules peuvent réaliser des fonctions égales (voir notamment l'exemple 6).

Définition. On dit que des formules $\mathfrak{A}$ et $\mathfrak{B}$ sur $\mathfrak{P}$ sont *équivalentes* si les fonctions qu'elles engendrent sont égales, c'est-à-dire si $f_{\mathfrak{A}} = f_{\mathfrak{B}}$. Ce qu'on notera $\mathfrak{A} = \mathfrak{B}$.

Exemple 6.

1) $0 = (x \& \bar{x})$,

2) $(\bar{x}_1 (x_2 + x_3)) = \overline{\{x_1 \vee [(x_2 \rightarrow x_3)(x_3 \rightarrow x_2)]\}}$,

3) $(x \rightarrow y) = (\bar{y} \rightarrow \bar{x})$.

Dressons la liste des équivalences (des identités) caractérisant les propriétés d'un ensemble de fonctions élémentaires (essentiellement de l'ensemble $\{0, 1, \bar{x}, (x_1 \& x_2), (x_1 \vee x_2)\}$).

Désignons par $(x_1 \circ x_2)$ l'une quelconque des fonctions $(x_1 \& x_2)$, $(x_1 \vee x_2)$, $(x_1 + x_2)$. L'essentiel, c'est que le symbole $\circ$ ait partout le même sens dans l'identité.

1. La fonction $(x_1 \circ x_2)$ est associative:

$$((x_1 \circ x_2) \circ x_3) = (x_1 \circ (x_2 \circ x_3)).$$

2. La fonction $(x_1 \circ x_2)$ est commutative:

$$(x_1 \circ x_2) = (x_2 \circ x_1).$$

3. La conjonction et la disjonction sont distributives:

$$((x_1 \vee x_2) \& x_3) = ((x_1 \& x_3) \vee (x_2 \& x_3)),$$
$$((x_1 \& x_2) \vee x_3) = ((x_1 \vee x_3) \& (x_2 \vee x_3)).$$

4. La négation, la conjonction et la disjonction sont reliées par les relations:

$$\bar{\bar{x}} = x, \quad \overline{(x_1 \& x_2)} = (\bar{x}_1 \vee \bar{x}_2), \quad \overline{(x_1 \vee x_2)} = (\bar{x}_1 \& \bar{x}_2).$$

5. La conjonction et la disjonction jouissent des propriétés suivantes:

$$(x \& x) = x, \quad (x \vee x) = x,$$
$$(x \& \bar{x}) = 0, \quad (x \vee \bar{x}) = 1,$$
$$(x \& 0) = 0, \quad (x \vee 0) = x,$$
$$(x \& 1) = x, \quad (x \vee 1) = 1.$$

Ces identités se vérifient sans peine par comparaison des fonctions correspondant au premier et au second membre.

Remarques. 1. Pour alléger l'écriture des formules nous conviendrons que l'opération $\&$ est *plus forte* que l'opération $\vee$, c'est-à-dire qu'en l'absence de parenthèses on effectue d'abord l'opération $\&$ et ensuite $\vee$. Exemple: $(x_1 \& x_2 \vee x_3)$ signifie que $((x_1 \& x_2) \vee x_3)$.

2. La fonction $(x_1 \circ x_2)$ étant associative, on peut au lieu des formules $((x_1 \circ x_2) \circ x_3)$ et $(x_1 \circ (x_2 \circ x_3))$ se servir de l'expression $(x_1 \circ x_2 \circ x_3)$ qui n'est pas une formule mais qui peut le devenir par l'introduction de parenthèses, ceci étant les propriétés fonctionnelles ne changent pas quelle que soit la place des parenthèses.

3. Dans les formules où la fonction extérieure est soit une conjonction, soit une disjonction, soit une addition modulo 2, soit une implication, soit une fonction de Sheffer, on omet les parenthèses extérieures et, par exemple, l'on écrit $x_1 \rightarrow x_2$ pour $(x_1 \rightarrow x_2)$; on omet de même les parenthèses dans les expressions surmontées d'une barre et, par exemple, l'on écrit $\overline{x_1 \rightarrow x_2}$ pour $\overline{(x_1 \rightarrow x_2)}$.

Dans la suite on se servira des notations suivantes:

$$\underset{i=1}{\overset{s}{\&}} x_i = x_1 \& x_2 \& \ldots \& x_s,$$

$$\bigvee_{i=1}^{s} x_i = x_1 \vee x_2 \vee \ldots \vee x_s.$$

Ces notations ont un sens également pour $s = 1$.

Formulons quelques règles découlant des numéros 2 et 5 de la liste des identités des fonctions élémentaires.

Si l'un des facteurs d'un produit logique est nul, alors ce produit l'est également.

Dans un produit logique d'au moins deux facteurs on peut supprimer tout facteur égal à 1.

Dans une somme logique d'au moins deux termes on peut supprimer tout terme égal à 0.

Si un terme d'une somme logique est égal à 1, *alors cette somme est aussi égale à* 1.

Dans la suite, compte tenu des remarques 1 et 2, on se servira non pas de formules mais d'expressions qui en diffèrent par l'absence de quelques parenthèses. Ces expressions seront parfois appelées formules.

Il est évident que *si* $\mathfrak{A}'$ *est une sous-formule de la formule* $\mathfrak{A}$ *et si l'on remplace l'une quelconque de ses entrées par une formule équivalente* $\mathfrak{B}'$, *la formule* $\mathfrak{A}$ *se transforme en une formule équivalente* $\mathfrak{B}$.

Ce principe combiné aux identités pour les fonctions élémentaires et à toutes les identités obtenues par substitution de formules quelconques aux variables x_1, x_2, x_3, nous permet d'effectuer des transformations équivalentes et d'obtenir ainsi de nouvelles identités.

Exemple 7.

$$x_1 \vee x_1x_2 = x_1 \cdot 1 \vee x_1x_2 = x_1(x_2 \vee \bar{x}_2) \vee x_1x_2 =$$
$$= x_1x_2 \vee x_1\bar{x}_2 \vee x_1x_2 = x_1x_2 \vee x_1x_2 \vee x_1\bar{x}_2 =$$
$$= x_1x_2 \vee x_1\bar{x}_2 = x_1(x_2 \vee \bar{x}_2) = x_1 \cdot 1 = x_1.$$

Donc, $x_1 \vee x_1x_2 = x_1$, c'est-à-dire qu'on obtient une règle d'*absorption du produit* x_1x_2 *par le facteur* x_1.

Il existe encore une méthode de déduction d'identités, qui repose sur le principe de dualité.

Définition. On appelle *fonction duale d'une fonction* $f(x_1, \ldots, x_n)$ une fonction $[f(x_1, \ldots, x_n)]^*$ égale à $\bar{f}(\bar{x}_1, \ldots, \bar{x}_n)$.

Le tableau de la fonction duale (pour un ordre donné des combinaisons des variables) se déduit du tableau de la fonction $f(x_1, \ldots, x_n)$ par interversion (c'est-à-dire en remplaçant 0 par 1 et 1 par 0) et retournement de la colonne de la fonction (voir tableau 6).

Tableau 6

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	$[f(x_1, x_2, x_3)]^*$
0	0	0	1	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	0

Comme $[f(x_{i_1}, \ldots, x_{i_n})]^* = \bar{f}(\bar{x}_{i_1}, \ldots, \bar{x}_{i_n})$, la fonction $[f(x_{i_1}, \ldots, x_{i_n})]^*$ définit la même application que la fonction $[f(x_1, \ldots, x_n)]^*$. Désignons cette application par f^*. Alors

$$[f(x_1, \ldots, x_n)]^* = f^*(x_1, \ldots, x_n).$$

On voit immédiatement que
la fonction 0 est duale de 1,
la fonction 1 est duale de 0,
la fonction x est duale de x,
la fonction $\bar{x}$ est duale de $\bar{x}$,
la fonction $x_1 \& x_2$ est duale de $x_1 \vee x_2$,
la fonction $x_1 \vee x_2$ est duale de $x_1 \& x_2$.

De la définition de la dualité il s'ensuit que

$$f^{**} = (f^*)^* = f,$$

c'est-à-dire que *la fonction f est duale de f^** (propriété de réciprocité).

Supposons que $f(x_1, \ldots, x_n)$ est représentée par une formule $\mathfrak{A}$. On demande la forme de la formule $\mathfrak{B}$ réalisant $f^*(x_1, \ldots, x_n)$.

Désignons par $x_1, \ldots, x_n$ les divers symboles des variables dans les ensembles

$$(x_{11}, \ldots, x_{1p_1}), \ldots, (x_{m1}, \ldots, x_{mp_m}).$$

Théorème 2. *Si*

$$\Phi(x_1, \ldots, x_n) = f(f_1(x_{11}, \ldots, x_{1p_1}), \ldots, f_m(x_{m1}, \ldots, x_{mp_m})),$$

alors

$$\Phi^*(x_1, \ldots, x_n) = f^*(f_1^*(x_{11}, \ldots, x_{1p_1}), \ldots, f_m^*(x_{m1}, \ldots, x_{mp_m})).$$

Démonstration.

$$\Phi^*(x_1, \ldots, x_n) = \overline{\Phi}(\overline{x}_1, \ldots, \overline{x}_n) =$$
$$= \overline{f}(f_1(\overline{x}_{11}, \ldots, \overline{x}_{1p_1}), \ldots, f_m(\overline{x}_{m1}, \ldots, \overline{x}_{mp_m})) =$$
$$= \overline{f}(\overline{\overline{f}}_1(\overline{x}_{11}, \ldots, \overline{x}_{1p_1}), \ldots, \overline{\overline{f}}_m(\overline{x}_{m1}, \ldots, \overline{x}_{mp_m})) =$$
$$= \overline{f}(\overline{f_1^*}(x_{11}, \ldots, x_{1p_1}), \ldots, \overline{f_m^*}(x_{m1}, \ldots, x_{mp_m})) =$$
$$= f^*(f_1^*(x_{11}, \ldots, x_{1p_1}), \ldots, f_m^*(x_{m1}, \ldots, x_{mp_m})).$$

Le théorème entraîne le

Principe de dualité. *Si une formule $\mathfrak{A} = S[f_1, \ldots, f_s]$ réalise une fonction $f(x_1, \ldots, x_n)$, alors la formule $S[f_1^*, \ldots, f_s^*]$, c'est-à-dire la formule obtenue à partir de $\mathfrak{A}$ par substitution des fonctions $f_1^*, \ldots, f_s^*$ respectivement aux fonctions $f_1, \ldots, f_s$, réalise la fonction $f^*(x_1, \ldots, x_n)$.*

Cette formule sera appelée *formule duale de* $\mathfrak{A}$ et notée $\mathfrak{A}^*$. Donc

$$\mathfrak{A}^* = S[f_1^*, \ldots, f_s^*].$$

Pour les formules sur l'ensemble $\mathfrak{P} = \{0, 1, \overline{x}, x_1 \& x_2, x_1 \vee x_2\}$, le principe de dualité peut être formulé comme suit: *pour obtenir la formule $\mathfrak{A}^*$ duale d'une formule $\mathfrak{A}$, il faut dans cette dernière remplacer partout*

$$0 \text{ par } 1,\ 1 \text{ par } 0,\ \& \text{ par } \vee,\ \vee \text{ par } \&.$$

Ou si $\mathfrak{A} = S[0, 1, \overline{x}, x_1 \& x_2, x_1 \vee x_2]$, *alors*

$$\mathfrak{A}^* = S[1, 0, \overline{x}, x_1 \vee x_2, x_1 \& x_2].$$

Exemple 8.

1) $\mathfrak{A}_1(x_1, x_2) = x_1 \& x_2$, $\mathfrak{A}_1^*(x_1, x_2) = x_1 \vee x_2$;

2) $\mathfrak{A}_2(x_1, x_2) = x_1 x_2 \vee \bar{x}_1 \bar{x}_2$, $\mathfrak{A}_2^*(x_1, x_2) = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2)$.

Du principe de dualité il résulte que si

$$\mathfrak{A}(x_1, \ldots, x_n) = \mathfrak{B}(x_1, \ldots, x_n),$$

alors

$$\mathfrak{A}^*(x_1, \ldots, x_n) = \mathfrak{B}^*(x_1, \ldots, x_n).$$

Exemple 9. L'identité $\overline{x_1 \& x_2} = \bar{x}_1 \vee \bar{x}_2$ entraîne l'identité $\overline{x_1 \vee x_2} = \bar{x}_1 \& \bar{x}_2$.

Le principe de dualité permet de réduire presque de moitié les opérations nécessitées par la déduction d'identités lors de l'étude des propriétés des fonctions élémentaires. On donnera plus bas d'autres applications du principe de dualité.

§ 4. Développement des fonctions booléennes suivant les variables. Forme canonique disjonctive

En parlant du langage des formules nous avons sciemment laissé de côté un problème relativement important. Si $\mathfrak{P}$ comprend un certain nombre de fonctions élémentaires, toute fonction booléenne peut-elle être représentée par une formule? Nous allons étudier ce problème dans ce qui suit.

Posons

$$x^\sigma = x\sigma \vee \bar{x}\bar{\sigma},$$

où σ est un paramètre égal soit à 0, soit à 1. Il est évident que

$$x^\sigma = \begin{cases} \bar{x} & \text{pour } \sigma = 0, \\ x & \text{pour } \sigma = 1. \end{cases}$$

Il est immédiat de voir que $x^\sigma = 1$ si et seulement si $x = \sigma$, c'est-à-dire que la valeur de la « base » est égale à celle de l'« exposant ».

Théorème 3 (de développement des fonctions suivant les variables). *Toute fonction booléenne $f(x_1, \ldots, x_n)$ peut être représentée pour tout m ($1 \leqslant m \leqslant n$) sous la forme suivante:*

$$f(x_1, \ldots, x_m, x_{m+1}, \ldots, x_n) =$$
$$= \bigvee_{(\sigma_1, \ldots, \sigma_m)} x_1^{\sigma_1} \& \ldots \& x_m^{\sigma_m} \& f(\sigma_1, \ldots, \sigma_m, x_{m+1}, \ldots, x_n), \quad (*)$$

où la disjonction est prise sur toutes les combinaisons des valeurs des variables $x_1, \ldots, x_m$.

Cette représentation s'appelle *développement de la fonction f suivant les m variables $x_1, \ldots, x_m$.*

Démonstration. Considérons une combinaison quelconque $(\alpha_1, \ldots, \alpha_n)$ des valeurs des variables et montrons que les deux membres de la relation (*) prennent la même valeur. Le premier membre est égal à $f(\alpha_1, \ldots, \alpha_n)$. Le second, à

$$\bigvee_{(\sigma_1, \ldots, \sigma_m)} \alpha_1^{\sigma_1} \& \ldots \& \alpha_m^{\sigma_m} \& f(\sigma_1, \ldots, \sigma_m, \alpha_{m+1}, \ldots, \alpha_n) =$$
$$= \alpha_1^{\alpha_1} \& \ldots \& \alpha_m^{\alpha_m} \& f(\alpha_1, \ldots, \alpha_m, \alpha_{m+1}, \ldots, \alpha_n) =$$
$$= f(\alpha_1, \ldots, \alpha_n).$$

Comme conséquences on obtient deux développements spéciaux.

1) *Développement suivant une variable*:

$$f(x_1, \ldots, x_{n-1}, x_n) =$$
$$= x_n \& f(x_1, \ldots, x_{n-1}, 1) \vee \bar{x}_n \& f(x_1, \ldots, x_{n-1}, 0).$$

Les fonctions $f(x_1, \ldots, x_{n-1}, 0)$ et $f(x_1, \ldots, x_{n-1}, 1)$ s'appellent *composantes du développement.* Ce développement présente de l'intérêt lorsque des propriétés des fonctions booléennes sont établies par récurrence.

2) *Développement suivant toutes les n variables*:

$$f(x_1, \ldots, x_n) = \bigvee_{(\sigma_1, \ldots, \sigma_n)} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n} \& f(\sigma_1, \ldots, \sigma_n).$$

Pour $f(x_1, \ldots, x_n) \not\equiv 0$, il devient

$$\bigvee_{(\sigma_1, \ldots, \sigma_n)} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n} \& f(\sigma_1, \ldots, \sigma_n) =$$
$$= \bigvee_{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

On obtient en définitive

$$f(x_1, \ldots, x_n) = \bigvee_{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}. \qquad (**)$$

Ce développement s'appelle *forme canonique disjonctive.*

Le théorème suivant a trait à cette notion.

Théorème 4. *Toute fonction booléenne peut être représentée par une formule au moyen de la négation, la conjonction et la disjonction.*

Démonstration. 1) Soit $f(x_1, \ldots, x_n) \equiv 0$. On a manifestement

$$f(x_1, \ldots, x_n) = x_1 \& \bar{x}_1.$$

2) Soit $f(x_1, \ldots, x_n) \not\equiv 0$. Représentons-la par sa forme canonique disjonctive:

$$f(x_1, \ldots, x_n) = \bigvee_{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

Donc, dans les deux cas la fonction f est représentée par une formule faisant intervenir la négation, la conjonction et la disjonction.

Ainsi, toute fonction booléenne peut être définie par une formule sur $\mathfrak{P}$, où $\mathfrak{P}$ est un ensemble constitué de trois fonctions : la négation, la conjonction et la disjonction.

Le théorème précédent est constructif, car il permet de construire pratiquement pour chaque fonction la formule qui la réalise (forme canonique disjonctive) : dans le tableau de la fonction $f(x_1, \ldots, x_n)$ ($f \not\equiv 0$) repérons toutes les lignes $(\sigma_1, \ldots, \sigma_n)$ pour lesquelles $f(\sigma_1, \ldots, \sigma_n) = 1$; pour chacune de ces lignes formons le produit logique

$$x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n},$$

et ensuite relions toutes les conjonctions obtenues par le symbole de la disjonction.

Exemple 10. 1) Trouver la forme canonique disjonctive de la fonction $x_1 \rightarrow x_2$.

Cette fonction est égale à 1 sur les trois combinaisons (0, 0), (0, 1) et (1, 1) (voir tableau 3). Donc

$$x_1 \rightarrow x_2 = x_1^0 \& x_2^0 \vee x_1^0 \& x_2^1 \vee x_1^1 \& x_2^1 = \bar{x}_1 \& \bar{x}_2 \vee \bar{x}_1 \& x_2 \vee x_1 \& x_2.$$

2) Trouver la forme canonique disjonctive de la fonction définie par le tableau 7.

On a

$$f(x_1, x_2, x_3) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3.$$

La forme canonique disjonctive est une expression de type $\Sigma\Pi$, c'est-à-dire la somme logique des produits $x_i^{\sigma_i}$. Il se pose la question de savoir si les fonctions booléennes sont justiciables d'une décomposition de type $\Pi\Sigma$. Nous allons voir que pour $f \not\equiv 1$ la réponse

Tableau 7

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

est oui. Ecrivons à cet effet la forme canonique disjonctive de la fonction f^* (il est évident que $f^* \not\equiv 0$) :

$$f^*(x_1, \ldots, x_n) = \bigvee_{\substack{(\sigma_1, \ldots, \sigma_n) \\ f^*(\sigma_1, \ldots, \sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

Le principe de dualité nous donne

$$f^{**}(x_1, \ldots, x_n) = \underset{\substack{(\sigma_1, \ldots, \sigma_n) \\ f^*(\sigma_1, \ldots, \sigma_n)=1}}{\&} (x_1^{\sigma_1} \vee \ldots \vee x_n^{\sigma_n}).$$

Le premier membre est égal à $f(x_1, \ldots, x_n)$, le second peut être transformé comme suit :

$$\underset{\substack{(\sigma_1, \ldots, \sigma_n) \\ f^*(\sigma_1, \ldots, \sigma_n)=1}}{\&} (x_1^{\sigma_1} \vee \ldots \vee x_n^{\sigma_n}) = \underset{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\bar{\sigma}_1, \ldots, \bar{\sigma}_n)=0}}{\&} (x_1^{\sigma_1} \vee \ldots \vee x_n^{\sigma_n}) =$$

$$= \underset{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=0}}{\&} (x_1^{\bar{\sigma}_1} \vee \ldots \vee x_n^{\bar{\sigma}_n}).$$

On obtient donc la décomposition

$$f(x_1, \ldots, x_n) = \underset{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=0}}{\&} (x_1^{\bar{\sigma}_1} \vee \ldots \vee x_n^{\bar{\sigma}_n}).$$

Cette expression s'appelle *forme canonique conjonctive.*

Exemple 11. 1) Trouver la forme canonique conjonctive de la fonction $x_1 \rightarrow x_2$.

On a

$$x_1 \rightarrow x_2 = x_1^{\bar{1}} \vee x_2^{\bar{0}} = \bar{x}_1 \vee x_2.$$

2) Trouver la forme canonique conjonctive de la fonction $f(x_1, x_2, x_3)$ définie par le tableau 7.

On a

$$f(x_1, x_2, x_3) =$$
$$= (x_1 \vee \bar{x}_2 \vee x_3)(x_1 \vee \bar{x}_2 \vee \bar{x}_3)(\bar{x}_1 \vee x_2 \vee x_3)(\bar{x}_1 \vee \bar{x}_2 \vee x_3).$$

Ainsi, pour se donner des fonctions booléennes on peut, outre les tableaux, se servir du langage des formules sur un ensemble de fonctions composé de la négation, de la conjonction et de la disjonction. Comme le langage des tableaux est par son volume à peu près équivalent à celui des formes canoniques disjonctives (resp. conjonctives), on peut affirmer que le langage des formules qui utilise la négation, la conjonction et la disjonction n'est pas plus mauvais que le langage des tableaux. On démontre le fait qu'il est bien meilleur que le langage des tableaux. Illustrons ceci sur l'exemple de la fonction

$$f(x_1, \ldots, x_{20}) = x_1 \vee \ldots \vee x_{20}.$$

Le second membre de cette formule compte 39 symboles (20 symboles de variables et 19 symboles $\vee$), le tableau de $f(x_1, \ldots, x_{20})$ contient 2^{20}, soit plus d'un million, de lignes.

§ 5. Complétude et fermeture

On a vu plus haut que toute fonction booléenne peut être représentée par une formule dans laquelle interviennent les fonctions $\bar{x}$, $x_1 \& x_2$ et $x_1 \vee x_2$.

Il se pose la question de savoir si l'existence de tels systèmes n'est pas le fait du hasard. Notre intention n'est pas de donner une réponse exhaustive à cette question, mais de montrer qu'il existe d'autres systèmes de fonctions élémentaires, doués de cette propriété.

Définition. On dit qu'un système de fonctions $\{f_1, f_2, \ldots, f_s, \ldots\}$ de P_2 est fonctionnellement *complet* si toute fonction booléenne peut être représentée par une formule composée des fonctions de ce système.

Considérons des exemples de systèmes complets.

1. L'ensemble P_2 de toutes les fonctions booléennes est un système complet.

2. L'ensemble $\mathfrak{P} = \{\bar{x}, x_1 \& x_2, x_1 \vee x_2\}$ est un système complet.

Il est évident que tous les systèmes ne sont pas complets. Exemple, le système $\mathfrak{P} = \{0, 1\}$. Le théorème suivant ramène le problème de la complétude d'un système à celle d'un autre.

Théorème 5. *Soient donnés deux systèmes de fonctions de* P_2:

$$\mathfrak{P} = \{f_1, f_2, \ldots\}, \qquad \text{(I)}$$

$$\mathfrak{Q} = \{g_1, g_2, \ldots\}. \qquad \text{(II)}$$

Si le système I *est complet et chacune de ses fonctions se représente par une expression composée de fonctions du système* II, *alors le système* II *est complet.*

Démonstration. Soit h une fonction arbitraire de P_2. Le système I étant complet, on peut exprimer h par une formule sur $\mathfrak{P}$, soit

$$h = S[f_1, f_2, \ldots, f_s, \ldots]$$

(nous avons noté entre crochets toutes les fonctions du système $\mathfrak{P}$, alors qu'en fait seul un nombre fini d'entre elles interviennent dans la formule). Par hypothèse

$$f_1 = S_1[g_1, g_2, \ldots],$$

$$f_2 = S_2[g_1, g_2, \ldots],$$

$$\cdots\cdots\cdots$$

Dans la formule $S[f_1, f_2, \ldots]$ on peut exclure les fonctions $f_1, f_2, \ldots$ en les remplaçant par des formules sur $\mathfrak{Q}$ *). On a :

$$S[f_1, f_2, \ldots] = S[S_1[g_1, g_2, \ldots], S_2[g_1, g_2, \ldots], \ldots].$$

La dernière expression définit une formule sur $\mathfrak{Q}$ de structure S'. On obtient

$$S[S_1[g_1, g_2, \ldots], S_2[g_1, g_2, \ldots], \ldots] = S'[g_1, g_2, \ldots]$$

ou en définitive

$$h = S'[g_1, g_2, \ldots],$$

c'est-à-dire que nous avons exprimé h par une formule sur $\mathfrak{Q}$, c.q.f.d.

Ce théorème nous permet d'établir la complétude d'autres systèmes et d'allonger ainsi la liste des systèmes complets.

3. Le système $\mathfrak{P} = \{\bar{x}, x_1 \& x_2\}$ est complet. Pour le prouver prenons le système 2 pour système I (voir théorème 5) et le système 3 pour système II et utilisons l'identité

$$x_1 \vee x_2 = \overline{\bar{x}_1 \& \bar{x}_2},$$

qui dérive de la propriété 4 des fonctions élémentaires.

4. Le système $\mathfrak{P} = \{\bar{x}, x_1 \vee x_2\}$ est complet.

Ceci se démontre comme dans l'exemple précédent ou encore à l'aide du principe de dualité.

5. Le système $\mathfrak{P} = \{x_1 \mid x_2\}$ est complet.

Pour prouver ceci, prenons le système 3 pour système I et le système 5 pour système II. Il est immédiat que

$$x_1 \mid x_1 = \bar{x}_1, \quad (x_1 \mid x_2) \mid (x_1 \mid x_2) = \overline{x_1 \mid x_2} = x_1 \& x_2.$$

6. Le système $\mathfrak{P} = \{0, 1, x_1x_2, x_1 + x_2 \ (\mathrm{mod.}\ 2)\}$ est complet.

En prenant pour système I le système 3 et pour système II le système 6, on obtient

$$x_1 + 1 = \bar{x}_1, \quad x_1x_2 = x_1 \& x_2.$$

La formule formée avec les constantes 0 et 1 et les fonctions x_1x_2 et $x_1 + x_2$ se transformant en un polynôme mod. 2 par suppression des parenthèses et quelques opérations algébriques simples, on obtient le théorème suivant dû à I. Gégalkine.

*) Ces changements ne peuvent pas être toujours effectués simultanément, car de nouveaux symboles $f_1, f_2, \ldots$ peuvent apparaître. Si par exemple

$$f_1 = x_1 \& x_2, \quad g_1 = x_1 \vee x_2, \quad g_2 = x_1 + x_2, \quad h = x_1 \& (x_2 \& x_3),$$

alors

$$f_1 = x_1 \& x_2 = x_1 + x_2 + (x_1 \vee x_2),$$

$$h = x_1 \& (x_2 \& x_3) = x_1 + (x_2 \& x_3) + (x_1 \vee (x_2 \& x_3)) =$$
$$= x_1 + x_2 + x_3 + (x_2 \vee x_3) + (x_1 \vee (x_2 + x_3 + (x_2 \vee x_3))).$$

Théorème 6. *Toute fonction de P_2 peut être exprimée par un polynôme* mod. 2 (*dit polynôme de Gégalkine*).

Calculons le nombre de polynômes de Gégalkine en $x_1, \ldots, x_n$, c'est-à-dire le nombre d'expressions de la forme

$$\sum_{(i_1, \ldots, i_s)} a_{i_1, \ldots, i_s} x_{i_1} \ldots x_{i_s}.$$

Le nombre de termes $x_{i_1} \ldots x_{i_s}$ est égal au nombre de sous-ensembles $(i_1, \ldots, i_s)$ de n nombres $(1, \ldots, n)$, soit à 2^n. Comme $a_{i_1, \ldots, i_s}$ est égal à 0 ou à 1, le nombre cherché de polynômes est égal à 2^{2^n}, c'est-à-dire au nombre de toutes les fonctions booléennes de ces mêmes variables. Une conséquence immédiate est l'*unicité de la représentation des fonctions par des polynômes de Gégalkine.*

Exemple 12. Représenter $x_1 \vee x_2$ par un polynôme de Gégalkine.

Cherchons l'expression de $x_1 \vee x_2$ sous forme d'un polynôme à coefficients indéterminés:

$$x_1 \vee x_2 = ax_1x_2 + bx_1 + cx_2 + d.$$

Pour $x_1 = x_2 = 0$, on a $0 = d$.
Pour $x_1 = 0$, $x_2 = 1$, on a $1 = c$.
Pour $x_1 = 1$, $x_2 = 0$, on a $1 = b$.
Pour $x_1 = x_2 = 1$ on a $1 = a + b + c$, c'est-à-dire que $a = 1$.
On obtient finalement $x_1 \vee x_2 = x_1x_2 + x_1 + x_2$.

On voit sur les exemples cités qu'il existe de nombreux systèmes complets. Chacun d'eux peut être adopté pour ensemble des fonctions élémentaires. Donc, on peut utiliser divers langages de formules pour la donnée des fonctions booléennes. Seule la nature du problème considéré peut nous guider sur le choix du langage.

La notion de complétude est étroitement liée à celle de fermeture et de classe fermée.

Définition. Soit $\mathfrak{M}$ un sous-ensemble de fonctions de P_2. On appelle *fermeture* de $\mathfrak{M}$ l'ensemble de toutes les fonctions booléennes représentables par des expressions booléennes formées avec les fonctions de $\mathfrak{M}$. La fermeture de $\mathfrak{M}$ se note $[\mathfrak{M}]$.

Exemple 13.

1) $\mathfrak{M} = P_2$. Il est évident que $[\mathfrak{M}] = P_2$.

2) $\mathfrak{M} = \{1, x_1 + x_2\}$. La fermeture de cet ensemble est la classe L de toutes les fonctions linéaires, c'est-à-dire des fonctions de la forme

$$f(x_1, \ldots, x_n) = c_0 + c_1x_1 + \ldots + c_nx_n \text{ (mod. 2)},$$

où $c_i = 0, 1$ $(i = 0, \ldots, n)$.

Signalons quelques propriétés de la fermeture:

1) $[\mathfrak{M}] \supseteq \mathfrak{M}$;

2) $[[\mathfrak{M}]] = [\mathfrak{M}]$;

3) si $\mathfrak{M}_1 \subseteq \mathfrak{M}_2$, alors $[\mathfrak{M}_1] \subseteq [\mathfrak{M}_2]$;

4) $[\mathfrak{M}_1 \cup \mathfrak{M}_2] \supseteq [\mathfrak{M}_1] \cup [\mathfrak{M}_2]$.

Définition. On dit qu'une classe (un ensemble) $\mathfrak{M}$ est fonctionnellement *fermée* si $[\mathfrak{M}] = \mathfrak{M}$.

Exemple 14.

1) La classe $\mathfrak{M} = P_2$ est une classe fermée.

2) La classe $\mathfrak{M} = \{1, x_1 + x_2\}$ n'est pas fermée.

3) La classe L est fermée, puisqu'une expression linéaire formée d'expressions linéaires est linéaire.

Il est immédiat de voir que toute classe $[\mathfrak{M}]$ est fermée. Ceci permet d'obtenir d'innombrables exemples de classes fermées.

On peut donner une autre définition de la complétude (une définition équivalente du reste à la première) en termes de fermeture et de classe fermée: $\mathfrak{M}$ est un système complet si $[\mathfrak{M}] = P_2$.

§ 6. Classes fermées fondamentales. Théorème de complétude

Nous entamons ce paragraphe par l'étude de certaines classes fermées très importantes de P_2.

1. Désignons par T_0 la classe de toutes les fonctions booléennes $f(x_1, \ldots, x_n)$ préservant la constante 0, c'est-à-dire des fonctions telles que

$$f(0, 0, \ldots, 0) = 0.$$

Il est immédiat de voir que les fonctions $0, x, x_1 \& x_2, x_1 \vee x_2, x_1 + x_2$ sont de la classe T_0 mais pas les fonctions $1, \bar{x}$.

Comme le tableau d'une fonction f de T_0 contient la valeur 0 dans sa première ligne, la classe T_0 comprend exactement $(1/2)\, 2^{2^n}$ fonctions booléennes des variables $x_1, \ldots, x_n$.

Montrons que T_0 est une classe fermée. Comme T_0 contient la fonction identique, pour prouver qu'elle est fermée il suffit de montrer que si $f, f_1, \ldots, f_m$ appartiennent à T_0, il en est de même de la fonction

$$\Phi = f(f_1, \ldots, f_m).$$

Ceci résulte de la chaîne d'égalités

$$\Phi(0, \ldots, 0) = f(f_1(0, \ldots, 0), \ldots, f_m(0, \ldots, 0)) = f(0, \ldots, 0) = 0.$$

2. Désignons par T_1 la classe de toutes les fonctions booléennes $f(x_1, \ldots, x_n)$ préservant la constante 1, c'est-à-dire des fonctions telles que

$$f(1, \ldots, 1) = 1.$$

Il est immédiat de voir que les fonctions 1, x, $x_1 \& x_2$, $x_1 \vee x_2$ appartiennent à la classe T_1, mais pas les fonctions 0, $\bar{x}$.

La classe T_1 étant composée de fonctions duales de celles de la classe T_0 (ou comme nous dirons encore, la classe T_1 étant duale de la classe T_0), les résultats relatifs à la classe T_0 se transposent immédiatement à la classe T_1. La classe T_1 comprend (1/2) 2^{2^n} fonctions des variables $x_1, \ldots, x_n$ et est une classe fermée.

3. Désignons par S la classe de toutes les fonctions autoduales, c'est-à-dire des fonctions f de P_2, telles que $f^* = f$.

Il est évident que les fonctions x et $\bar{x}$ sont autoduales. Un exemple moins trivial de fonction autoduale est la fonction $h(x_1, x_2, x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3$:

$$h^*(x_1, x_2, x_3) = (x_1 \vee x_2)(x_1 \vee x_3)(x_2 \vee x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3 = h(x_1, x_2, x_3).$$

Toute fonction autoduale est telle que

$$\bar{f}(\bar{x}_1, \ldots, \bar{x}_n) = f(x_1, \ldots, x_n);$$

autrement dit, une fonction autoduale prend des valeurs opposées sur les combinaisons $(\alpha_1, \ldots, \alpha_n)$ et $(\bar{\alpha}_1, \ldots, \bar{\alpha}_n)$ que nous appellerons *opposées*. Il s'ensuit qu'une fonction autoduale est complètement définie par ses valeurs sur la première moitié de lignes (voir tableau 1). Donc, le nombre de fonctions autoduales des variables $x_1, \ldots, x_n$ est $2^{2^{n-1}} = \sqrt{2^{2^n}}$.

Montrons maintenant que la classe S est fermée. Comme cette classe contient la fonction identique, il suffit de montrer que la fonction

$$\Phi = f(f_1, \ldots, f_m)$$

est autoduale si $f, f_1, \ldots, f_m$ le sont. La dernière proposition est immédiate:

$$\Phi^* = f^*(f_1^*, \ldots, f_m^*) = f(f_1, \ldots, f_m) = \Phi.$$

Prouvons maintenant un lemme de non-autodualité des fonctions.

Lemme 1. *Si $f(x_1, \ldots, x_n) \notin S$, en y substituant x et $\bar{x}$ on peut obtenir une fonction non autoduale d'une seule variable, c'est-à-dire une constante.*

Démonstration. Comme $f \notin S$, il existe une combinaison $(\alpha_1, \ldots, \alpha_n)$ telle que

$$f(\bar{\alpha}_1, \ldots, \bar{\alpha}_n) = f(\alpha_1, \ldots, \alpha_n).$$

Considérons les fonctions $\varphi_i(x) = x^{\alpha_i}$ $(i = 1, \ldots, n)$ et posons

$$\varphi(x) = f(\varphi_1(x), \ldots, \varphi_n(x)).$$

On a

$$\varphi(0) = f(\varphi_1(0), \ldots, \varphi_n(0)) = f(0^{\alpha_1}, \ldots, 0^{\alpha_n}) = f(\bar{\alpha}_1, \ldots, \bar{\alpha}_n) =$$
$$= f(\alpha_1, \ldots, \alpha_n) = f(1^{\alpha_1}, \ldots, 1^{\alpha_n}) =$$
$$= f(\varphi_1(1), \ldots, \varphi_n(1)) = \varphi(1),$$

ce qui prouve le lemme.

4. On se servira des notations vectorielles suivantes:

$$\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n),$$
$$\tilde{\beta} = (\beta_1, \ldots, \beta_n),$$
$$f(\tilde{\alpha}) = f(\alpha_1, \ldots, \alpha_n), \text{ etc.}$$

Définition. On dit que les combinaisons $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ et $\tilde{\beta} = (\beta_1, \ldots, \beta_n)$ vérifient la *relation d'antériorité* $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ si

$$\alpha_1 \leqslant \beta_1, \ldots, \alpha_n \leqslant \beta_n.$$

Exemple: $(0, 1, 0, 1) \preccurlyeq (1, 1, 0, 1)$.

Il est évident que si $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ et $\tilde{\beta} \preccurlyeq \tilde{\gamma}$, alors $\tilde{\alpha} \preccurlyeq \tilde{\gamma}$. A signaler que la relation d'antériorité n'est pas caractéristique de tous les couples de combinaisons. Les combinaisons (0, 1) et (1, 0) ne vérifient pas cette relation. Donc, l'ensemble de toutes les combinaisons de longueur n est partiellement ordonné par la relation d'antériorité $\preccurlyeq$.

Définition. On dit qu'une fonction $f(x_1, \ldots, x_n)$ est *monotone* si pour deux combinaisons quelconques $\tilde{\alpha}$ et $\tilde{\beta}$ telles que $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ on a

$$f(\tilde{\alpha}) \leqslant f(\tilde{\beta}).$$

Les fonctions $0, 1, x, x_1 \& x_2, x_1 \vee x_2$ sont visiblement des fonctions monotones.

Désignons par M l'ensemble de toutes les fonctions monotones. Montrons que M est une classe fermée. Pour le prouver, il suffit, puisque M contient la fonction identique, de montrer que

$$\Phi = f(f_1, \ldots, f_m)$$

est une fonction monotone si $f, f_1, \ldots, f_m$ le sont. Soient

$$\tilde{x} = (x_1, \ldots, x_n), \quad \tilde{x}^1 = (x_{11}, \ldots, x_{1p_1}), \ldots, \tilde{x}^m = (x_{m1}, \ldots, x_{mp_m})$$

des combinaisons de variables des fonctions $\Phi, f_1, \ldots, f_m$, l'ensemble des variables de la fonction Φ n'étant composé que des

variables dont dépendent les fonctions $f_1, \ldots, f_m$. Soient $\tilde{\alpha}$ et $\tilde{\beta}$ deux combinaisons de longueur n des valeurs des variables $\tilde{x}$, $\tilde{\alpha} \preccurlyeq \tilde{\beta}$. Ces combinaisons définissent des combinaisons $\tilde{\alpha}^1, \tilde{\beta}^1, \ldots \ldots, \tilde{\alpha}^m, \tilde{\beta}^m$ de valeurs des variables $\tilde{x}^1, \ldots, \tilde{x}^m$ telles que $\tilde{\alpha}^1 \preccurlyeq \tilde{\beta}^1, \ldots, \tilde{\alpha}^m \preccurlyeq \tilde{\beta}^m$. Comme les fonctions $f_1, \ldots, f_m$ sont monotones on a

$$f_1(\tilde{\alpha}^1) \leqslant f_1(\tilde{\beta}^1), \ldots, f_m(\tilde{\alpha}^m) \leqslant f_m(\tilde{\beta}^m).$$

Donc

$$(f_1(\tilde{\alpha}^1), \ldots, f_m(\tilde{\alpha}^m)) \preccurlyeq (f_1(\tilde{\beta}^1), \ldots, f_m(\tilde{\beta}^m)),$$

et, puisque f est monotone,

$$f(f_1(\tilde{\alpha}^1), \ldots, f_m(\tilde{\alpha}^m)) \leqslant f(f_1(\tilde{\beta}^1), \ldots, f_m(\tilde{\beta}^m)),$$

d'où

$$\Phi(\tilde{\alpha}) = f(f_1(\tilde{\alpha}^1), \ldots, f_m(\tilde{\alpha}^m)) \leqslant f(f_1(\tilde{\beta}^1), \ldots, f_m(\tilde{\beta}^m)) = \Phi(\tilde{\beta}).$$

On dira que des combinaisons $\tilde{\alpha}$ et $\tilde{\beta}$ sont *adjacentes* ou *voisines* (par rapport à la i-ème coordonnée) si

$$\tilde{\alpha} = (\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_n),$$
$$\tilde{\beta} = (\alpha_1, \ldots, \alpha_{i-1}, \bar{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_n).$$

Montrons un lemme de non-monotonie des fonctions.

Lemme 2. *Si $f(x_1, \ldots, x_n) \notin M$, en y substituant les constantes* 0 *et* 1 *et la fonction x, on peut en déduire la fonction $\bar{x}$.*

Démonstration. Montrons tout d'abord qu'il existe un couple de combinaisons adjacentes $\tilde{\alpha}$ et $\tilde{\beta}$ telles que $\tilde{\alpha} \preccurlyeq \tilde{\beta}$ et

$$f(\tilde{\alpha}) > f(\tilde{\beta}).$$

En effet, comme $f \notin M$ il existe des combinaisons $\tilde{\alpha}^1$ et $\tilde{\beta}^1$ telles que $\tilde{\alpha}^1 \preccurlyeq \tilde{\beta}^1$ et $f(\tilde{\alpha}^1) > f(\tilde{\beta}^1)$. Si les combinaisons $\tilde{\alpha}^1$ et $\tilde{\beta}^1$ sont adjacentes, on obtient ce qu'on voulait. Si $\tilde{\alpha}^1$ et $\tilde{\beta}^1$ ne sont pas adjacentes, alors $\tilde{\beta}^1$ diffère de $\tilde{\alpha}^1$ en t coordonnées, où $t > 1$, et de plus ces t coordonnées prennent la valeur 0 dans la combinaison $\tilde{\alpha}^1$ et la valeur 1 dans la combinaison $\tilde{\beta}^1$. Par suite, entre $\tilde{\alpha}^1$ et $\tilde{\beta}^1$ on peut insérer $t-1$ combinaisons intermédiaires $\tilde{\alpha}^2$,

$\tilde{\alpha}^3, \ldots, \tilde{\alpha}^t$ telles que

$$\tilde{\alpha}^1 \preccurlyeq \tilde{\alpha}^2 \preccurlyeq \ldots \preccurlyeq \tilde{\alpha}^t \preccurlyeq \tilde{\beta}^1.$$

Il est évident que les combinaisons consécutives sont adjacentes. Comme $f(\tilde{\alpha}^1) > f(\tilde{\beta}^1)$, alors pour l'un au moins de ces couples de combinaisons, que nous désignerons par $\tilde{\alpha}$ et $\tilde{\beta}$ $(\tilde{\alpha} \preccurlyeq \tilde{\beta})$, on aura $f(\tilde{\alpha}) > f(\tilde{\beta})$. Supposons que ces combinaisons sont adjacentes par rapport à la i-ème coordonnée et par suite

$$\tilde{\alpha} = (\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n),$$
$$\tilde{\beta} = (\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n).$$

Considérons la fonction

$$\varphi(x) = f(\alpha_1, \ldots, \alpha_{i-1}, x, \alpha_{i+1}, \ldots, \alpha_n).$$

On a

$$\varphi(0) = f(\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n) =$$
$$= f(\tilde{\alpha}) > f(\tilde{\beta}) = f(\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n) = \varphi(1).$$

Ce qui signifie que $\varphi(0) = 1$ et $\varphi(1) = 0$, c'est-à-dire que $\varphi(x) = \bar{x}$, c.q.f.d.

5. Voyons enfin la classe L des fonctions linéaires.

Cette classe contient visiblement les constantes 0, 1, la fonction identique x, les fonctions $\bar{x}$, $x_1 + x_2$, mais pas les fonctions $x_1 \mathbin{\&} x_2$ et $x_1 \vee x_2$. On a vu plus haut que cette classe est également fermée.

Prouvons un lemme relatif à une fonction non linéaire.

Lemme 3. *Si $f(x_1, \ldots, x_n) \notin L$, en y substituant les constantes* 0 *et* 1 *ainsi que les fonctions $\bar{x}$ et x et éventuellement en barrant f, on peut déduire la fonction $x_1 \mathbin{\&} x_2$.*

Démonstration. Considérons le polynôme de Gégalkine pour f:

$$f(x_1, \ldots, x_n) = \sum_{(i_1, \ldots, i_s)} a_{i_1 \ldots i_s} x_{i_1} \ldots x_{i_s}.$$

Le polynôme n'étant pas linéaire, il contient un terme qui est le produit d'au moins deux facteurs. Sans nuire à la généralité on peut considérer que x_1 et x_2 figurent parmi ces facteurs. On peut alors transformer ce polynôme de la façon suivante:

$$\sum_{(i_1, \ldots, i_s)} a_{i_1 \ldots i_s} x_{i_1} \ldots x_{i_s} = x_1 x_2 f_1(x_3, \ldots, x_n) +$$
$$+ x_1 f_2(x_3, \ldots, x_n) + x_2 f_3(x_3, \ldots, x_n) + f_4(x_3, \ldots, x_n),$$

où $f_1(x_3, \ldots, x_n) \not\equiv 0$ en vertu de l'unicité du polynôme.

Supposons que $\alpha_3, \ldots, \alpha_n$ sont telles que $f_1(\alpha_3, \ldots, \alpha_n) = 1$. Alors

$$\varphi(x_1, x_2) = f(x_1, x_2, \alpha_3, \ldots, \alpha_n) = x_1x_2 + \alpha x_1 + \beta x_2 + \gamma,$$

où α, β et γ sont des constantes égales à 0 ou à 1. Considérons la fonction $\psi(x_1, x_2)$ déduite à partir de $\varphi(x_1, x_2)$ de la manière suivante:

$$\psi(x_1, x_2) = \varphi(x_1 + \beta, x_2 + \alpha) + \alpha\beta + \gamma.$$

Il est évident que

$$\varphi(x_1 + \beta, x_2 + \alpha) + \alpha\beta + \gamma =$$
$$= (x_1 + \beta)(x_2 + \alpha) + \alpha(x_1 + \beta) + \beta(x_2 + \alpha) + \gamma + \alpha\beta + \gamma =$$
$$= x_1x_2.$$

Donc

$$\psi(x_1, x_2) = x_1 \,\&\, x_2.$$

Ce qui achève la démonstration du lemme.

Remarquons en conclusion que les classes fermées T_0, T_1, S, M et L sont deux à deux distinctes, ce qui ressort du tableau 8 dans

Tableau 8

	T_0	T_1	S	M	L
0	+	—	—	+	+
1	—	+	—	+	+
$\bar{x}$	—	—	+	—	+

lequel le signe + indique que la fonction appartient à la classe mentionnée et le signe —, non.

Passons maintenant à l'examen de l'un des plus importants problèmes de l'algèbre logique, le problème des conditions nécessaire et suffisante de complétude. Soit

$$\mathfrak{P} = \{f_1, f_2, \ldots, f_s, \ldots\}$$

un système de fonctions de P_2. Le théorème suivant nous permet de dire si ce système est complet ou non.

Théorème 7 (de complétude fonctionnelle). *Pour que le système de fonctions $\mathfrak{P}$ soit complet il est nécessaire et suffisant qu'il ne soit entièrement contenu dans aucune des cinq classes fermées T_0, T_1, S, M et L.*

Démonstration. *Nécessité.* Supposons que $\mathfrak{P}$ est complet, c'est-à-dire que $[\mathfrak{P}] = P_2$, et qu'il est contenu dans l'une des classes

indiquées, que nous noterons $\mathfrak{N}$, i.e. $\mathfrak{P} \subseteq \mathfrak{N}$. Alors en vertu des propriétés de fermeture et de la fermeture de $\mathfrak{N}$, on a

$$P_2 = [\mathfrak{P}] \subseteq [\mathfrak{N}] = \mathfrak{N}.$$

Donc $\mathfrak{N} = P_2$. Cette contradiction prouve la condition nécessaire.

Suffisance. Supposons que $\mathfrak{P}$ n'est entièrement contenu dans aucune des cinq classes mentionnées. On peut alors extraire de $\mathfrak{P}$ un sous-système $\mathfrak{P}'$ composé d'au plus cinq fonctions et qui jouira de la même propriété. A cet effet, prenons dans $\mathfrak{P}$ des fonctions f_i, f_j, f_k, f_m et f_l n'appartenant respectivement pas aux classes T_0, T_1, S, M et L et posons

$$\mathfrak{P}' = \{f_i, f_j, f_k, f_l, f_m\}.$$

On peut admettre que ces fonctions dépendent des mêmes variables $x_1, \ldots, x_n$ (voir remarque du § 1).

Nous ferons la démonstration de la suffisance en trois étapes.

I. Construction des constantes 0 et 1 à l'aide des fonctions f_i, f_j et f_k.

Considérons la fonction $f_i \notin T_0$. Deux cas sont possibles:

1. $f_i(1, \ldots, 1) = 1$. Alors $\varphi(x) = f_i(x, \ldots, x)$ vaut 1 puisque

$$\varphi(0) = f_i(0, \ldots, 0) = 1, \quad \varphi(1) = f_i(1, \ldots, 1) = 1.$$

La constante 0 s'obtient à partir de f_j: $f_j(1, \ldots, 1) = 0$.

2. $f_i(1, \ldots, 1) = 0$. Alors $\varphi(x) = f_i(x, \ldots, x)$ est égale à $\bar{x}$, car

$$\varphi(0) = f_i(0, \ldots, 0) = 1, \quad \varphi(1) = f_i(1, \ldots, 1) = 0.$$

Considérons f_k ($f_k \notin S$). Comme nous avons $\bar{x}$, en vertu du lemme 1, on peut obtenir une constante à partir de f_k. On peut de même à l'aide de $\bar{x}$ obtenir la deuxième constante. Donc, dans les deux cas, on obtient les constantes 0 et 1.

II. Construction de la fonction $\bar{x}$ à l'aide des constantes 0, 1 et de la fonction f_m. Elle repose sur le lemme 2.

III. Construction de la fonction $x_1 \& x_2$ à l'aide des constantes 0, 1 et des fonctions $\bar{x}$ et f_l. Elle repose sur le lemme 3.

On a donc réalisé les fonctions $\bar{x}$ et $x_1 \& x_2$ à l'aide de formules sur $\mathfrak{P}'$ (donc sur $\mathfrak{P}$). Ceci prouve la suffisance.

Corollaire 1. *Toute classe fermée $\mathfrak{M}$ de fonctions de P_2 telle que $\mathfrak{M} \neq P_2$ est contenue dans l'une au moins des classes construites.*

Définition. On dit qu'une classe $\mathfrak{N}$ de fonctions de P_2 est *précomplète* (ou *maximale*) si $\mathfrak{N}$ n'est pas complète et si pour toute fonction f ($f \in P_2$, $f \notin \mathfrak{N}$) la classe $\mathfrak{N} \cup \{f\}$ est complète.

De la définition il s'ensuit qu'une classe précomplète est fermée.

Corollaire 2. *En algèbre logique seules les classes T_0, T_1, S, M et L sont précomplètes.*

Considérons un exemple illustrant les possibilités du théorème de complétude fonctionnelle.

Exemple 15. Montrons que le système de fonctions

$$f_1 = x_1x_2, \quad f_2 = 0, \quad f_3 = 1 \quad \text{et} \quad f_4 = x_1 + x_2 + x_3 \ (\text{mod. } 2)$$

est complet.

On a: $f_3 \notin T_0$, $f_2 \notin T_1$, $f_2 \notin S$, $f_4 \notin M$, $f_1 \in L$. D'autre part, l'élimination de l'une quelconque de ces fonctions nous conduit au système non complet:

$$\{f_2, f_3, f_4\} \subset L, \quad \{f_1, f_3, f_4\} \subset T_1,$$
$$\{f_1, f_2, f_4\} \subset T_0, \quad \{f_1, f_2, f_3\} \subset M.$$

Une conséquence immédiate de la démonstration du théorème 7 est le

Théorème 8. *De tout système $\mathfrak{P}$ complet dans P_2, on peut extraire un sous-système complet contenant quatre fonctions au plus.*

Démonstration. On a vu que de $\mathfrak{P}$ on peut extraire un sous-système complet $\mathfrak{P}'$ contenant au plus cinq fonctions. Il s'avère par ailleurs que la fonction $f_i \notin T_0$ soit n'est pas autoduale (cas 1) puisque $f_i(0, \ldots, 0) = f_i(1, \ldots, 1)$, soit n'est pas monotone (cas 2) puisque $f_i(0, \ldots, 0) > f_i(1, \ldots, 1)$. Donc, le système $\{f_i, f_j, f_m, f_l\}$ ou le système $\{f_i, f_k, f_l\}$ sera complet.

L'exemple 15 montre que la constante 4 ne peut être diminuée.

Le théorème de complétude fonctionnelle ne fait pas que donner un critère de complétude, il permet (combiné à la représentation par la forme canonique conjonctive ou disjonctive) de trouver pour une fonction booléenne quelconque f une expression composée de fonctions du système complet $\mathfrak{P}$.

§ 7. Aperçu des résultats de Post

Le mathématicien américain E. Post a apporté une profonde contribution à l'étude des classes fermées de P_2. Il a décrit la structure de toutes les classes fermées de P_2. Formulons certains des résultats les plus saillants.

Définition. On dit qu'un système de fonctions $\{f_1, f_2, \ldots, f_s, \ldots\}$ appartenant à une classe fermée $\mathfrak{M}$ est *complet dans* $\mathfrak{M}$ si sa fermeture est confondue avec $\mathfrak{M}$.

Autrement dit, un système est complet dans $\mathfrak{M}$ si toute fonction de $\mathfrak{M}$ peut être exprimée par des fonctions de ce système.

Définition. On dit qu'un système de fonctions $\{f_1, f_2, \ldots, f_s, \ldots\}$ appartenant à une classe fermée $\mathfrak{M}$ est une *base* de $\mathfrak{M}$ s'il est complet dans $\mathfrak{M}$ et si aucun de ses sous-systèmes propres ne l'est dans $\mathfrak{M}$.

Ainsi, le système

$$f_1 = x_1 x_2, \quad f_2 = 0, \quad f_3 = 1, \quad f_4 = x_1 + x_2 + x_3$$

est une base dans P_2. On montre que le système de fonctions $\{0, 1, x_1 \,\&\, x_2, x_1 \vee x_2\}$ est une base dans la classe M.

Théorème 9 (Post [21]). *Toute classe fermée de P_2 possède une base finie.*

Théorème 10 (Post [21]). *L'ensemble des classes fermées de P_2 est dénombrable.*

Post a établi le deuxième théorème et ensuite le premier, bien que le deuxième soit une conséquence logique du premier.

CHAPITRE 2

LOGIQUE k-VALENTE

Les logiques k-valentes s'introduisent comme une généralisation de la logique binaire. De ce fait notre exposé sera bref par endroit et l'on passera sur quelques définitions et démonstrations identiques. Attirons l'attention sur les deux circonstances suivantes:

1) de nombreuses propriétés et résultats de la logique binaire s'étendent aux logiques k-valentes;

2) certains faits des logiques k-valentes présentent des distinctions fondamentales avec l'algèbre logique.

Pour ces raisons, certains problèmes ne possèdent pas une solution aussi exhaustive qu'en algèbre logique, d'autres ne sont pas résolus du tout.

§ 8. Fonctions de la logique k-valente. Formules et réalisation de fonctions par des formules

Soit $U = \{u_1, u_2, \ldots, u_m, \ldots\}$ un alphabet de variables (d'arguments). On étudiera des fonctions $f(u_{i_1}, \ldots, u_{i_n})$ ($u_{i_\nu} \neq u_{i_\mu}$ pour $\nu \neq \mu$) dont les variables sont définies sur l'ensemble $E^k = \{0, 1, \ldots, k-1\}$, telles que $f(\alpha_1, \ldots, \alpha_n) \in E^k$ pour $\alpha_i \in E^k$ ($i = 1, 2, \ldots, n$).

Pour simplifier l'écriture on se servira pour les variables de U des métasymboles $x, y, z, \ldots$ et $x_i, y_i, z_i, \ldots$ et pour les fonctions, de la notation plus simple

$$f(x_1, \ldots, x_n).$$

Il est évident que la fonction $f(x_1, \ldots, x_n)$ est entièrement définie par la donnée de son tableau (voir tableau 9). Dans ce tableau, les combinaisons sont les représentations des nombres $0, 1, \ldots, k^n - 1$ dans un système de numération à base k. Le symbole f sera interprété comme celui d'une application caractérisée par un tableau, les symboles $x_1, x_2, \ldots, x_n$, comme les noms des colonnes. Les fonctions

Tableau 9

$x_1 \ldots x_{n-1} \; x_n$	$f(x_1, \ldots, x_{n-1}, x_n)$
$0 \ldots 0 \; 0$	$f(0, \ldots, 0, 0)$
$0 \ldots 0 \; 1$	$f(0, \ldots, 0, 1)$
. . . .	. . . .
$0 \ldots 0 \; k-1$	$f(0, \ldots, 0, k-1)$
$0 \ldots 1 \; 0$	$f(0, \ldots, 1, 0)$
. . . .	. . . .
$k-1 \ldots k-1 \; k-1$	$f(k-1, \ldots, k-1, k-1)$

d'une variable seront représentées par des tableaux ou par la substitution (généralisée)

$$S(x) = \begin{pmatrix} 0 & 1 & \ldots & k-1 \\ i_0 & i_1 & \ldots & i_{k-1} \end{pmatrix},$$

où $S(\alpha) = i_\alpha$.

Désignons par P_k l'ensemble formé de toutes les fonctions de la logique k-valente sur l'alphabet U, et des constantes $0, 1, \ldots, k-1$. Le nombre des combinaisons $(\alpha_1, \ldots, \alpha_n)$ des valeurs des variables $x_1, \ldots, x_n$ étant égal à k^n, on a le résultat suivant.

Théorème 1. *Le nombre des fonctions de P_k dépendant des n variables $x_1, \ldots, x_n$ est égal à k^{k^n}.*

Il s'ensuit de ce qui précède que dans P_k pour $k \geqslant 3$ on se heurte à de plus grandes difficultés que dans P_2 aussi bien quand il s'agit de tirer le meilleur parti de la donnée tabulaire des fonctions que d'analyser toutes les fonctions de n variables. Dans P_3 déjà, le nombre de fonctions de deux variables est égal à $3^9 = 19683$, c'est-à-dire que cet ensemble est démesuré. Aussi, dans P_k la donnée tabulaire des fonctions est-elle souvent délaissée au profit d'un algorithme de calcul. Par exemple,

$$\max(x_1, \ldots, x_n)$$

peut être considéré comme un algorithme qui délivre le maximum de toute combinaison $(\alpha_1, \ldots, \alpha_n)$ des valeurs des variables. Cet algorithme définit dans P_k une fonction unique que nous désignerons par le même symbole.

On introduit ensuite, comme dans P_2, les notions de variables essentielles et inessentielles, et d'égalité des fonctions. Ceci nous permet de considérer les fonctions dans P_k aux variables inessentielles près.

On traite ensuite des exemples de fonctions de P_k qui seront considérées comme des fonctions « élémentaires ».

1) $\bar{x} = x + 1$ (mod. k). La fonction $\bar{x}$ est ici une généralisation de la négation au sens d'une translation « cyclique » des valeurs.

2) $Nx = k - 1 - x$. La fonction Nx que l'on notera encore $\sim x$ est une autre généralisation de la négation qui en littérature a reçu le nom de *négation de Lukasiewicz.*

3) $I_i(x) = \begin{cases} k-1 & \text{pour } x = i, \\ 0 & \text{pour } x \neq i \end{cases} \quad (i = 0, \ldots, k-1).$

Pour $i \neq k - 1$ les fonctions $I_i(x)$ généralisent certaines propriétés de la négation.

4) $j_i(x) = \begin{cases} 1 & \text{pour } x = i, \\ 0 & \text{pour } x \neq i. \end{cases}$

La fonction $j_i(x)$ est la fonction caractéristique de la valeur i et pour $i \neq k - 1$ est une généralisation de la négation.

5) $\min(x_1, x_2)$, une généralisation de la conjonction.

6) x_1x_2 (mod. k), une autre généralisation de la conjonction.

7) $\max(x_1, x_2)$, une généralisation de la disjonction.

8) $x_1 + x_2$ (mod. k).

De l'examen de cette liste de fonctions élémentaires, il ressort que les fonctions booléennes admettent en logique k-valente ($k \geqslant 3$) plusieurs analogues dont chacun d'eux généralise la propriété correspondante de la fonction.

Ensuite, de même qu'en algèbre logique, on introduit la notion de formule sur un ensemble de fonctions $\mathfrak{P}$. On désignera les formules par les symboles $\mathfrak{A}$, $\mathfrak{B}$, . . . avec ou sans indices. Si l'on veut spécifier la dépendance d'une formule par rapport à des variables ou mettre en évidence les fonctions qui composent la formule, on se servira des notations

$$\mathfrak{A}(x_1, \ldots, x_n), \quad \mathfrak{A}[f_1, \ldots, f_s].$$

A toute formule $\mathfrak{A}(x_1, \ldots, x_n)$ est associée une fonction $f(x_1, \ldots, x_n)$ de P_k. Cela étant, on dira aussi que la formule $\mathfrak{A}$ réalise ou engendre ou représente la fonction f. La superposition de fonctions de $\mathfrak{P}$ et l'opération de superposition ont ici le même sens. On introduit ensuite la notion d'équivalence des formules $\mathfrak{A}$ et $\mathfrak{B}$: $\mathfrak{A} = \mathfrak{B}$ si les fonctions $f_{\mathfrak{A}}$ et $f_{\mathfrak{B}}$ engendrées par $\mathfrak{A}$ et $\mathfrak{B}$ sont égales.

La notion d'équivalence nous permet de décrire les principales propriétés des fonctions élémentaires. Indiquons quelques-unes d'entre elles. Supposons que $(x_1 \circ x_2)$ désigne l'une quelconque des fonctions $\min(x_1, x_2)$, x_1x_2 (mod. k), $\max(x_1, x_2)$, $x_1 + x_2$ (mod. k).

1. La fonction $(x_1 \circ x_2)$ est associative:

$$((x_1 \circ x_2) \circ x_3) = (x_1 \circ (x_2 \circ x_3)).$$

2. La fonction $(x_1 \circ x_2)$ est commutative:

$$(x_1 \circ x_2) = (x_2 \circ x_1).$$

On désignera parfois les fonctions min (x_1, x_2) et max (x_1, x_2) respectivement par $(x_1 \& x_2)$ et $(x_1 \vee x_2)$. Compte tenu de l'associativité et du fait que l'opération & prime sur l'opération $\vee$ (voir remarques 1, 2 et 3, § 3, chap. 1) on peut, pour noter les formules, se servir des expressions obtenues par suppression de certaines parenthèses.

Indiquons encore quelques groupes d'identités (règles) relatives au système de fonctions élémentaires

$$\{0, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}.$$

3. Règles de descente du symbole I à l'« intérieur » d'une formule:

$$I_\sigma(c) = \begin{cases} k-1 & \text{pour } c = \sigma, \\ 0 & \text{pour } c \neq \sigma \end{cases} \quad (\sigma, c = 0, 1, \ldots, k-1);$$

$$I_\sigma(I_\tau(x)) = \begin{cases} I_0(x) \vee \ldots \vee I_{\tau-1}(x) \vee I_{\tau+1}(x) \vee \ldots \vee I_{k-1}(x) \text{ pour } \sigma = 0, \\ 0 \text{ pour } 0 < \sigma < k-1, \\ I_\tau(x) \text{ pour } \sigma = k-1; \end{cases}$$

$$I_\sigma(x_1 x_2) = I_\sigma(x_1)(I_\sigma(x_2) \vee \ldots \vee I_{k-1}(x_2)) \vee$$
$$\vee I_\sigma(x_2)(I_0(x_1) \vee \ldots \vee I_{k-1}(x_1));$$

$$I_\sigma(x_1 \vee x_2) = I_\sigma(x_1)(I_0(x_2) \vee \ldots \vee I_\sigma(x_2)) \vee$$
$$\vee I_\sigma(x_2)(I_0(x_1) \vee \ldots \vee I_\sigma(x_1)).$$

4. Propriété de distributivité:

$$(x_1 \vee x_2) x_3 = (x_1 x_3) \vee (x_2 x_3),$$
$$(x_1 x_2) \vee x_3 = (x_1 \vee x_3)(x_2 \vee x_3).$$

5. Règle d'élimination des entrées « pures » d'une variable:

$$x = 1 \cdot I_1(x) \vee 2 \cdot I_2(x) \vee \ldots \vee (k-1) I_{k-1}(x).$$

6. Règle d'introduction d'une variable:

$$x_1 = x_1 (I_0(x_2) \vee \ldots \vee I_{k-1}(x_2)).$$

7. Règles de simplification:

$$I_\sigma(x) I_\tau(x) = \begin{cases} I_\sigma(x) & \text{pour } \tau = \sigma, \\ 0 & \text{pour } \tau \neq \sigma; \end{cases}$$

$$\sigma\tau = \min(\sigma, \tau); \quad \sigma \vee \tau = \max(\sigma, \tau);$$
$$(k-1) x = x; \quad 0 \cdot x = 0;$$
$$(k-1) \vee x = k-1; \quad 0 \vee x = x.$$

Par des transformations équivalentes on peut obtenir de nouvelles identités à partir de quelques identités (par exemple les identités 1 à 7) relatives au système de fonctions élémentaires

$$\{0,\ 1,\ \ldots,\ k-1,\ I_0(x),\ \ldots,\ I_{k-1}(x),\ \min(x_1,\ x_2),\ \max(x_1,\ x_2)\}.$$

L'étude des propriétés des fonctions élémentaires montre que ces propriétés ne caractérisent pas toutes les généralisations des fonctions booléennes. Illustrons ceci sur des exemples.

Exemple 1.

1) $\sim(\sim x) = x$, mais $\bar{\bar{x}} \neq x$ (pour $k \geqslant 3$).

2) $\sim\min(x_1,\ x_2) = \max(\sim x_1,\ \sim x_2)$, mais

$$\overline{\min(x_1,\ x_2)} \neq \max(\bar{x}_1,\ \bar{x}_2).$$

En conclusion nous allons exhiber une identité analogue à la forme canonique disjonctive qui jouera un rôle important dans la suite de l'exposé:

$$f(x_1,\ \ldots,\ x_n) = \bigvee_{(\sigma_1,\ \ldots,\ \sigma_n)} I_{\sigma_1}(x_1)\ \&\ \ldots\ \&\ I_{\sigma_n}(x_n)\ \&\ f(\sigma_1,\ \ldots,\ \sigma_n).$$

La démonstration se réduit à une vérification directe.

Il est aisé de voir que toute formule sur l'ensemble de fonctions élémentaires $\{0,\ 1,\ \ldots,\ k-1,\ I_0(x),\ \ldots,\ I_{k-1}(x),\ \min(x_1,\ x_2),\ \max(x_1,\ x_2)\}$ peut être transformée en une forme normale disjonctive à l'aide des identités 1 à 7. De là on peut démontrer aisément que les règles 1 à 7 permettent de passer d'une formule quelconque sur l'ensemble donné à une formule équivalente. Ceci montre que le système d'identités 1 à 7 est dans un certain sens complet.

§ 9. Exemples de systèmes complets

Dans P_k la définition de la complétude est la même que dans P_2.

Définition. On dit qu'un système $\mathfrak{P}$ de fonctions $f_1, f_2, \ldots, f_s, \ldots$ de P_k est fonctionnellement *complet* si toute fonction de P_k peut être représentée par une formule sur ce système.

On verra plus bas des exemples de systèmes complets. Pour prouver la complétude d'un système on ramènera ce problème à celui de la complétude d'un autre système (voir § 5, chap. 1).

1. Le système $\mathfrak{P} = P_k$ est complet. Il est évident que l'ensemble de toutes les fonctions de P_k est complet.

2. Le système

$$\mathfrak{P} = \{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2).\ \max(x_1, x_2)\}$$

est complet.

Théorème 2. *Le système*

$$\mathfrak{P} = \{0, \ldots, k-1, \ I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}$$

est complet dans P_k.

Démonstration. Soit $f(x_1, \ldots, x_n)$ une fonction arbitraire de P_k. Elle admet la décomposition

$$f(x_1, \ldots, x_n) = \bigvee_{(\sigma_1, \ldots, \sigma_n)} I_{\sigma_1}(x_1) \,\&\, \ldots \,\&\, I_{\sigma_n}(x_n) \,\&\, f(\sigma_1, \ldots, \sigma_n).$$

Cette décomposition ne fait intervenir que des fonctions de $\mathfrak{P}$. C.q.f.d.

3. Le système $\mathfrak{P} = \{\bar{x}, \max(x_1, x_2)\}$ est complet.

Théorème 3. *Le système* $\mathfrak{P} = \{\bar{x}, \max(x_1, x_2)\}$ *est complet dans* P_k.

Démonstration. a) *Construction des constantes.* La fonction $\bar{x} = x + 1$ nous donne les fonctions

$$x + 2 = (x + 1) + 1, \ \ldots, \ x + k - 1 = (x + k - 2) + 1,$$

$$x = x + k = (x + (k - 1)) + 1.$$

On voit immédiatement que

$$\max(x, x + 1, \ldots, x + k - 1) = k - 1.$$

De là on obtient les autres constantes à l'aide de $\bar{x}$.

b) *Construction des fonctions d'une variable.* On obtient d'abord les fonctions $I_i(x)$ $(i = 0, \ldots, k - 1)$:

$$I_i(x) = 1 + \max_{\alpha \neq k-1-i} \{x + \alpha\}.$$

En effet, si $x = i$, le premier membre est égal à $k - 1$ et le second à

$$1 + \max_{\alpha \neq k-1-i} \{i + \alpha\} = 1 + \max_{i+\alpha \neq k-1} \{i + \alpha\} = 1 + k - 2 = k - 1;$$

si $x \neq i$, le premier membre est égal à 0 et le second à

$$1 + \max_{\alpha \neq k-1-i} \{x + \alpha\} = 1 + (x + (k - 1 - x)) = k = 0.$$

Introduisons la fonction

$$f_{s,i}(x) = \begin{cases} s \text{ pour } x = i, \\ 0 \text{ pour } x \neq i. \end{cases}$$

Montrons que

$$f_{s,i}(x) = s + 1 + \max[I_i(x), k - 1 - s].$$

Ceci ressort facilement des graphiques représentés sur la figure **1**.

Si $f(x)$ est une fonction arbitraire d'une variable de P_k, alors

$$f(x) = \max \{f_{f(0),\,0}(x),\ f_{f(1),\,1}(x),\ \ldots,\ f_{f(k-1),\,k-1}(x)\},$$

et en particulier

$$\sim x = \max \{f_{k-1,\,0}(x),\ f_{k-2,\,1}(x),\ \ldots,\ f_{0,\,k-1}(x)\}.$$

c) *Déduction de* $\min(x_1, x_2)$. On a vu que

$$\sim \min(x_1, x_2) = \max(\sim x_1, \sim x_2),$$

d'où

$$\min(x_1, x_2) = \sim \max(\sim x_1, \sim x_2).$$

Donc toute fonction du système

$$\{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\},$$

dont on a prouvé la complétude, peut être exprimée par des formules sur le système $\mathfrak{P}$. Donc, le système $\mathfrak{P}$ est complet. C.q.f.d.

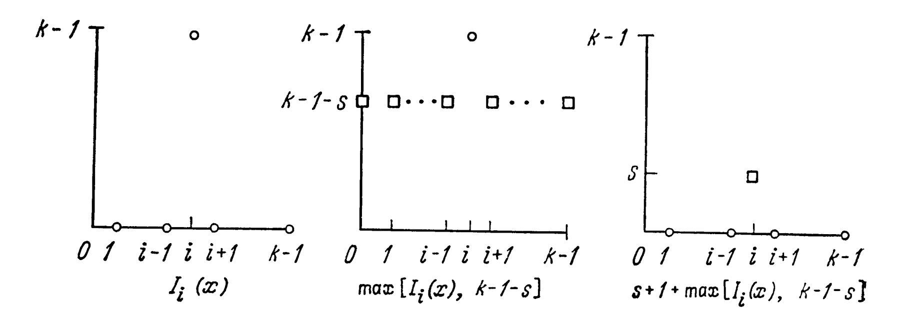


Fig. 1

Posons: $V_k(x_1, x_2) = \max(x_1, x_2) + 1 \pmod{k}$. La fonction $V_k(x_1, x_2)$ s'appelle *fonction de Webb*. C'est un analogue de la fonction de Sheffer.

4. Le système $\mathfrak{P} = \{V_k(x_1, x_2)\}$ est complet. Le problème de cette complétude peut être ramené à celui de la complétude du système 3.

A la notion de complétude est reliée la notion de fermeture et de classe fermée.

Définition. Soit $\mathfrak{M}$ un sous-ensemble quelconque de fonctions de P_k. On appelle *fermeture* de $\mathfrak{M}$ l'ensemble $[\mathfrak{M}]$ de toutes les fonctions de P_k représentables par des formules sur $\mathfrak{M}$.

Définition. On dit qu'une classe (un ensemble) $\mathfrak{M}$ est fonctionnellement *fermée* si $[\mathfrak{M}] = \mathfrak{M}$.

Tout ce qui a été dit sur ces notions dans P_2 se généralise à P_k. Voici des exemples d'ensembles fermés.

Exemple 2. 1) La classe $\mathfrak{M} = P_k$ est visiblement fermée.

2) Soit $\mathcal{E} \subset E^k$. Désignons par $T_{\mathcal{E}}$ l'ensemble de toutes les fonctions $f(x_1, \ldots, x_n)$ de P_k telles que $f(\alpha_1, \ldots, \alpha_n) \in \mathcal{E}$ si $\alpha_i \in \mathcal{E}$ $(i = 1, \ldots, n)$. En d'autres termes $T_{\mathcal{E}}$ est l'ensemble de toutes les fonctions de P_k *conservant* $\mathcal{E}$; on notera ceci par

$$f(\mathcal{E}, \mathcal{E}, \ldots, \mathcal{E}) \subseteq \mathcal{E}.$$

La classe $\mathfrak{M} = T_{\mathcal{E}}$ est manifestement fermée.

On peut donner de la complétude une définition en termes de fermeture qui est équivalente à la première : $\mathfrak{M}$ est un système complet si $[\mathfrak{M}] = P_k$.

La notion de classe fermée peut être appliquée à l'établissement de la non-complétude de certains systèmes.

Exemple 3. Considérons le système $\mathfrak{P} = \{\sim x, \max(x_1, x_2)\}$. Soit $\mathcal{E} = \{0, k-1\}$. Comme ces deux fonctions conservent $\mathcal{E}$, on a

$$[\mathfrak{P}] \subseteq T_{\mathcal{E}}.$$

Comme $\mathcal{E} \neq E^k$ pour $k \geqslant 3$, $T_{\mathcal{E}}$ ne contient pas, par exemple, la constante 1. Donc, $\mathfrak{P}$ ne sera pas un système complet pour $k \geqslant 3$. Sur cet exemple on voit que le système $\{\sim x, \max(x_1, x_2)\}$ n'est pas complet bien qu'il soit une généralisation du système $\{\bar{x}, x_1 \vee x_2\}$ de fonctions booléennes.

§ 10. Reconnaissance de la complétude. Théorème de complétude

Passons maintenant à l'étude de la complétude de systèmes arbitraires $\mathfrak{P}$. On s'occupera donc du problème de savoir comment d'après un ensemble $\mathfrak{P}$ on peut dire s'il est complet ou non. Ce problème admet deux approches.

La première approche est algorithmique. Elle implique de préciser ce qu'on entend par l'expression « soit donné un système arbitraire $\mathfrak{P}$ ». A cet effet, nous allons restreindre ce problème et supposer que le système est fini :

$$\mathfrak{P} = \{f_1, \ldots, f_s\},$$

donc il peut être donné explicitement soit par un tableau soit par une liste de formules. Comme pour $k = 2$, on peut admettre que chaque fonction du système $\mathfrak{P}$ dépend des variables $x_1, x_2, \ldots, x_n$. Le problème peut s'énoncer sous la forme suivante : existe-t-il un algorithme permettant de dire si un système $\mathfrak{P}$ fini est complet ou non (*problème de reconnaissance de la complétude*).

Pour $p \geqslant 1$ quelconque, posons

$$g_i^p(x_1, \ldots, x_p) = x_i,$$

et soit $\mathfrak{M}_{x_1, \ldots, x_p}$ l'ensemble de toutes les fonctions de $\mathfrak{M}$ dépendant des variables $x_1, \ldots, x_p$.

Théorème 4. *Il existe un algorithme de reconnaissance de la complétude.*

Démonstration. Construisons par récurrence une suite d'ensembles

$$\mathfrak{N}_0, \mathfrak{N}_1, \ldots, \mathfrak{N}_r, \ldots$$

de fonctions de deux variables x_1 et x_2.

Base de la récurrence. Posons $\mathfrak{N}_0 = \Lambda$, où Λ est un ensemble vide.

Passage de la récurrence. Supposons qu'on ait déjà formé les ensembles $\mathfrak{N}_0, \mathfrak{N}_1, \ldots, \mathfrak{N}_r$; montrons comment il faut définir l'ensemble $\mathfrak{N}_{r+1}$. Ecrivons à cet effet les fonctions qui composent $\mathfrak{N}_r$ $(r \geqslant 0)$:

$$\mathfrak{N}_r = \{h_1(x_1, x_2), \ldots, h_{s_r}(x_1, x_2)\} \quad (s_r = 0 \text{ pour } r = 0),$$

et considérons pour chaque i $(i = 1, \ldots, s)$ toutes les formules de la forme

$$f_i(H_1(x_1, x_2), \ldots, H_n(x_1, x_2)),$$

où $H_l(x_1, x_2)$ est soit la fonction $h_j(x_1, x_2)$ $(j = 1, \ldots, s_r)$, soit $g_\sigma^2(x_1, x_2)$.

Ainsi, en examinant les $s(s_r + 2)^n$ formules, on obtiendra peut-être des fonctions n'appartenant pas à $\mathfrak{N}_r$. Désignons-les par

$$h_{s_r+1}(x_1, x_2), \ldots, h_{s_{r+1}}(x_1, x_2).$$

Posons $\mathfrak{N}_{r+1} = \mathfrak{N}_r \cup \{h_{s_r+1}(x_1, x_2), \ldots, h_{s_{r+1}}(x_1, x_2)\}$. Il est évident que

$$\mathfrak{N}_0 \subseteq \mathfrak{N}_1 \subseteq \ldots \subseteq \mathfrak{N}_r \subseteq \ldots$$

De la construction il est clair que si $\mathfrak{N}_{r+1} = \mathfrak{N}_r$, alors $\mathfrak{N}_r = \mathfrak{N}_{r+1} = \ldots$, c'est-à-dire que la suite des ensembles « se stabilise ». Désignons par r^* le plus petit indice à partir duquel a lieu cette « stabilisation ». Alors la suite d'ensembles

$$\mathfrak{N}_0 \subset \mathfrak{N}_1 \subset \ldots \subset \mathfrak{N}_{r^*}$$

sera strictement croissante. Comme la puissance de $\mathfrak{N}_i$ est $\leqslant k^{k^2}$, on a $r^* \leqslant k^{k^2}$. Donc, la « stabilisation » peut intervenir au bout d'un nombre fini de pas. Considérons l'ensemble $\mathfrak{N}_{r^*}$. Deux cas sont possibles.

1) $\mathfrak{N}_{r^*}$ contient toutes les fonctions des variables x_1 et x_2, donc $V_k(x_1, x_2)$. Le système donné est alors complet.

2) $\mathfrak{N}_{r^*}$ ne contient pas toutes les fonctions de x_1 et de x_2. Comme $[\mathfrak{P}]_{x_1 x_2} = \mathfrak{N}_{r^*}$, alors $[\mathfrak{P}]$ ne contient pas toutes les fonctions de x_1 et de x_2. Par suite, $\mathfrak{P}$ n'est pas complet.

L'algorithme consiste donc à construire les classes $\mathfrak{N}_0$, $\mathfrak{N}_1$, ..., $\mathfrak{N}_{r*}$ jusqu'à la « stabilisation » et à définir ensuite par l'étude de la classe $\mathfrak{N}_{r*}$ si le système est complet ou non. Ceci achève la démonstration du théorème.

Théorème 5. *De tout système $\mathfrak{P}$ complet dans P_k on peut extraire un sous-système fini qui est aussi complet.*

Démonstration. Soit $\mathfrak{P} = \{f_1, f_2, \ldots, f_s, \ldots\}$. Le système étant complet, la fonction $V_k(x_1, x_2)$ peut être exprimée au moyen des fonctions du système $\mathfrak{P}$ par une formule:

$$V_k(x_1, x_2) = \mathfrak{A}[f_{i_1}, \ldots, f_{i_r}].$$

Il est évident que le sous-système $\{f_{i_1}, \ldots, f_{i_r}\}$ est le système cherché. C.q.f.d.

Donc, il n'existe pas de systèmes complets essentiellement infinis et la restriction imposée dans le problème de reconnaissance de la complétude n'est pas aussi forte qu'elle ne pouvait le paraître de prime abord.

La deuxième approche implique la vérification de certaines propriétés de la classe $\mathfrak{P}$.

Nous allons introduire une notion et prouver deux lemmes relatifs à cette notion.

Soit $\mathfrak{N}$ la classe des fonctions $h_j(y_1, \ldots, y_p)$ de P_k dépendant de p variables $y_1, \ldots, y_p$. Supposons qu'elle contient les fonctions $g_i(y_1, \ldots, y_p) = y_i$ $(i = 1, \ldots, p)$.

Définition. On dit qu'une fonction $f(x_1, \ldots, x_n)$ *préserve l'ensemble* $\mathfrak{N}$ si pour des fonctions quelconques $h_{i_1}(y_1, \ldots, y_p), \ldots \ldots, h_{i_n}(y_1, \ldots, y_p)$ de $\mathfrak{N}$ on a

$$f(h_{i_1}(y_1, \ldots, y_p), \ldots, h_{i_n}(y_1, \ldots y_p)) \in \mathfrak{N}.$$

Désignons par $\mathfrak{M}$ la classe de toutes les fonctions de P_k préservant $\mathfrak{N}$.

Exemple 4. Soient $k = 2$, $p = 1$ et $\mathfrak{N} = \{y, \bar{y}\}$. Alors la fonction $f(x_1, \ldots, x_n)$ qui préserve l'ensemble $\mathfrak{N}$ vérifie la relation

$$f(y^{\sigma_1}, \ldots, y^{\sigma_n}) = y^{\sigma},$$

d'où

$$f(\bar{\sigma}_1, \ldots, \bar{\sigma}_n) = \bar{f}(\sigma_1, \ldots, \sigma_n).$$

Donc la fonction $f(x_1, \ldots, x_n)$ est autoduale et la classe des fonctions préservant $\mathfrak{N}$ est la classe S des fonctions autoduales.

Lemme 1. *La classe $\mathfrak{M}$ des fonctions préservant $\mathfrak{N}$ est fermée.*

Démonstration. Il est évident que la classe $\mathfrak{M}$ contient la fonction identique. Donc, pour prouver que la classe $\mathfrak{M}$ est fermée il

suffit de montrer que la fonction $\Phi = f(f_1, \ldots, f_m)$ est un élément de $\mathfrak{M}$ si les fonctions $f, f_1, \ldots, f_m$ le sont. Supposons que Φ dépend de n variables. Soient $h_{i_1}, \ldots, h_{i_n}$ des fonctions arbitraires de $\mathfrak{N}$. Alors

$$\Phi(h_{i_1}, \ldots, h_{i_n}) = f(f_1(h_{i_1}, \ldots, h_{i_n}), \ldots, f_m(h_{i_1}, \ldots, h_{i_n})) =$$
$$= f(H_1, \ldots, H_m),$$

où les fonctions $H_1, \ldots, H_m$ appartiennent à la classe $\mathfrak{N}$, donc $f(H_1, \ldots, H_m)$ appartient aussi à $\mathfrak{N}$. Ceci prouve le lemme.

Lemme 2. *Si la classe $\mathfrak{N}$ est telle que $[\mathfrak{N}]_{y_1 \ldots y_p} = \mathfrak{N}$, alors la classe $\mathfrak{M}$ des fonctions préservant $\mathfrak{N}$ est telle que*

$$\mathfrak{M}_{y_1 \ldots y_p} = \mathfrak{N}.$$

Démonstration. Soit $h(y_1, \ldots, y_p) \in \mathfrak{N}$. Alors

$$h(h_{i_1}, \ldots, h_{i_p}) \in [\mathfrak{N}]_{y_1 \ldots y_p} = \mathfrak{N},$$

c'est-à-dire que $h \in \mathfrak{M}_{y_1 \ldots y_p}$. D'autre part, si

$$f(y_1, \ldots, y_p) \in \mathfrak{M}_{y_1 \ldots y_p},$$

en substituant $g_1, \ldots, g_p$ à $y_1, \ldots, y_p$, on obtient

$$f(g_1, \ldots, g_p) \in \mathfrak{N} \text{ ou } f(y_1, \ldots, y_p) \in \mathfrak{N}.$$

Ce qui prouve le lemme.

Théorème 6 (de complétude fonctionnelle de A. Kouznetsov [5]). *On peut construire un système de classes fermées dans P_k*

$$\mathfrak{M}_1, \mathfrak{M}_2, \ldots, \mathfrak{M}_s$$

dont aucune n'est entièrement contenue dans une autre classe, et tel qu'un sous-système de fonctions de P_k est complet si et seulement s'il n'est entièrement contenu dans aucune des classes $\mathfrak{M}_1, \ldots, \mathfrak{M}_s$.

Démonstration. *Construction du système de classes $\mathfrak{M}_1, \ldots, \mathfrak{M}_s$.* Soit $\mathfrak{N}_1, \mathfrak{N}_2, \ldots, \mathfrak{N}_l$ le système de tous les sous-ensembles propres de fonctions de P_k dépendant des variables x_1 et x_2, qui vérifient les conditions suivantes ($i = 1, \ldots, l$):

1) $\mathfrak{N}_i$ contient les fonctions $g_1(x_1, x_2) = x_1$ et $g_2(x_1, x_2) = x_2$;

2) $[\mathfrak{N}_i]_{x_1 x_2} = \mathfrak{N}_i$.

On construit les sous-ensembles indiqués en examinant tous les sous-ensembles propres de l'ensemble des fonctions de P_k dépendant de x_1 et x_2 (dont le nombre est $< 2^{k^{k^2}}$). Cela étant, on ne retient que les sous-ensembles qui contiennent les deux fonctions g_1 et g_2. Ensuite on vérifie la condition $[\mathfrak{N}]_{x_1 x_2} = \mathfrak{N}$ pour tout sous-ensemble retenu comme dans le théorème de reconnaissance de la complétude.

Désignons par $\mathfrak{M}'_i$ la classe des fonctions préservant le sous-ensemble $\mathfrak{N}_i$. Les lemmes 1 et 2 nous disent que $\mathfrak{M}'_i$ est une classe fermée telle que

$$[\mathfrak{M}'_i]_{x_1x_2} = \mathfrak{N}_i$$

(c'est-à-dire que toutes les classes sont distinctes).

Il reste ensuite à éliminer les classes qui sont contenues dans une autre. On obtient en définitive le système

$$\mathfrak{M}_1, \mathfrak{M}_2, \ldots, \mathfrak{M}_s.$$

Nécessité. Elle résulte de la fermeture et de la non-complétude des classes $\mathfrak{M}_i$.

Suffisance. Soit $\mathfrak{M} \subset P_k$ un système de fonctions qui n'est entièrement contenu dans aucune des classes $\mathfrak{M}_i$ $(i = 1, \ldots, s)$. Admettons que $\mathfrak{M}$ est une classe fermée. Désignons par $\mathfrak{M}'$ la classe $[\mathfrak{M} \cup \{g_1, g_2\}]$. Il est évident que les classes $\mathfrak{M}$ et $\mathfrak{M}'$ sont simultanément complètes ou non complètes, puisque

$$\mathfrak{M}' = \mathfrak{M} \cup [\{g_1, g_2\}]$$

et que la fonction $V_k(x_1, x_2)$ soit appartient simultanément à $\mathfrak{M}$ et à $\mathfrak{M}'$, soit n'appartient à aucune de ces classes. Prenons $\mathfrak{N}' = \mathfrak{M}'_{x_1x_2}$. Montrons que $\mathfrak{N}'$ contient toutes les fonctions de x_1 et de x_2. En effet, dans le cas contraire il est évident que

$$\mathfrak{N}' \equiv \mathfrak{N}_i \text{ et } \mathfrak{M}' \subseteq \mathfrak{M}'_i \subseteq \mathfrak{M}_j.$$

Comme $\mathfrak{M} \subseteq \mathfrak{M}'$, il vient que $\mathfrak{M} \subseteq \mathfrak{M}_j$, ce qui est absurde. Donc, $\mathfrak{N}'$, et par suite $\mathfrak{M}'$, contient la fonction $V_k(x_1, x_2)$. Il s'ensuit que la classe $\mathfrak{M}'$, donc la classe $\mathfrak{M}$, est complète. C.q.f.d.

Attirons l'attention sur le fait que le théorème de Kouznetsov prouve qu'il est possible d'exprimer les conditions de complétude du système $\mathfrak{P}$ en termes d'appartenance de ce système à des classes spéciales $\mathfrak{M}_1, \ldots, \mathfrak{M}_s$. Cependant, la construction de ces classes même pour des k peu élevés implique des calculs insurmontables. D'où la nécessité de chercher des critères plus efficaces. Nous verrons que cèla est possible moyennant des restrictions, c'est-à-dire moyennant la connaissance d'une information subsidiaire sur le système $\mathfrak{P}$.

§ 11. Quelques propriétés des fonctions essentielles. Critère de complétude

Dans ce paragraphe on se propose de prouver un critère de complétude. Pour cela on aura besoin d'étudier plus en détail les propriétés des fonctions $f(x_1, \ldots, x_n)$ de P_k qui dépendent essentiellement de deux variables au moins. Ces fonctions seront appelées *essentielles*. Prouvons trois lemmes.

Lemme 3. *Soit $f(x_1, \ldots, x_n)$ une fonction essentielle prenant l $(l \geqslant 3)$ valeurs. Soit x_1 une variable essentielle. Il existe alors deux combinaisons $(\alpha, \alpha_2, \ldots, \alpha_n)$ et $(\beta, \alpha_2, \ldots, \alpha_n)$ telles que*

$$f(\alpha, \alpha_2, \ldots, \alpha_n) \neq f(\beta, \alpha_2, \ldots, \alpha_n)$$

et $f(\alpha, x_2, \ldots, x_n)$ prend une valeur différente de $f(\alpha, \alpha_2, \ldots, \alpha_n)$ et de $f(\beta, \alpha_2, \ldots, \alpha_n)$.

Démonstration. Comme x_1 est une variable essentielle de la fonction f, il existe des valeurs $\alpha_2, \ldots, \alpha_n$ telles que la suite

$$f(0, \alpha_2, \ldots, \alpha_n), \quad f(1, \alpha_2, \ldots, \alpha_n), \ldots, f(k-1, \alpha_2, \ldots, \alpha_n) \quad (*)$$

contienne au moins deux valeurs différentes. Deux cas sont possibles.

1) La suite $(*)$ ne contient pas toutes les l valeurs. Considérons une combinaison $(\alpha, \gamma_2, \ldots, \gamma_n)$ telle que la valeur $f(\alpha, \gamma_2, \ldots, \gamma_n)$ n'appartienne pas à la suite $(*)$. Il est évident que

$$f(\alpha, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \gamma_2, \ldots, \gamma_n).$$

Soit β l'une quelconque des valeurs pour lesquelles

$$f(\beta, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \alpha_2, \ldots, \alpha_n).$$

Il est évident que

$$f(\beta, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \gamma_2, \ldots, \gamma_n).$$

2) La suite $(*)$ contient toutes les l valeurs. La fonction f étant essentielle, il existe un α tel que

$$f(\alpha, x_2, \ldots, x_n) \not\equiv \text{const}$$

(sinon f ne dépendrait essentiellement que d'une variable). Il s'ensuit qu'il existe des combinaisons $(\alpha, \alpha_2, \ldots, \alpha_n)$ et $(\alpha, \gamma_2, \ldots, \gamma_n)$ telles que

$$f(\alpha, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \gamma_2, \ldots, \gamma_n).$$

Comme la suite $(*)$ contient l $(l \geqslant 3)$ valeurs, il existe un β tel que

$$f(\beta, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \alpha_2, \ldots, \alpha_n),$$
$$f(\beta, \alpha_2, \ldots, \alpha_n) \neq f(\alpha, \gamma_2, \ldots, \gamma_n).$$

Ce qui prouve le lemme.

Soit G un ensemble fini. Désignons par $|G|$ le cardinal de G.

Lemme 4 (fondamental). *Si $f(x_1, \ldots, x_n)$ est une fonction essentielle prenant au moins l $(l \geqslant 3)$ valeurs, alors il existe n sous-ensembles $G_1, \ldots, G_n$ de l'ensemble E^k tels que*

$$1 \leqslant |G_1|, \ldots, |G_n| \leqslant l - 1$$

et que la fonction f prend l valeurs sur l'ensemble des combinaisons $(\alpha_1, \ldots, \alpha_n)$, où $\alpha_i \in G_i$ $(i = 1, \ldots, n)$, c'est-à-dire que sur $G_1 \times \times G_2 \times \ldots \times G_n$ la fonction f prend l valeurs.

Démonstration. On peut admettre que x_1 est une variable essentielle de la fonction f. D'après le lemme précédent il existe trois combinaisons

$$(\alpha, \alpha_2, \ldots, \alpha_n), (\beta, \alpha_2, \ldots, \alpha_n), (\alpha, \gamma_2, \ldots, \gamma_n)$$

sur lesquelles f prend trois valeurs distinctes. Ajoutons à ces trois combinaisons les $l - 3$ combinaisons

$$(\delta_1^{(1)}, \delta_2^{(1)}, \ldots, \delta_n^{(1)}), \ldots, (\delta_1^{(l-3)}, \delta_2^{(l-3)}, \ldots, \delta_n^{(l-3)}),$$

sur lesquelles f prend les $l - 3$ valeurs restantes.

Posons

$$G_1 = \{\alpha, \beta, \delta_1^{(1)}, \ldots, \delta_1^{(l-3)}\},$$
$$G_2 = \{\alpha_2, \gamma_2, \delta_2^{(1)}, \ldots, \delta_2^{(l-3)}\},$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$G_n = \{\alpha_n, \gamma_n, \delta_n^{(1)}, \ldots, \delta_n^{(l-3)}\}.$$

Les ensembles construits vérifient visiblement les hypothèses du lemme. C.q.f.d.

Définition. On dit qu'un système de la forme

$$(\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \alpha_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \beta_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \alpha_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \beta_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta_j, \alpha_{j+1}, \ldots, \alpha_n)$$

est un *carré* si $\alpha_i \neq \beta_i$ et $\alpha_j \neq \beta_j$.

Lemme 5. *Soit $f(x_1, \ldots, x_n)$ une fonction essentielle prenant l valeurs $(l \geqslant 3)$. Il existe alors un carré sur lequel f prend soit plus de deux valeurs, soit deux valeurs, dont l'une en un seul point de ce carré.*

Démonstration. Admettons que x_1 est une variable essentielle de la fonction f. En vertu du lemme 1 il existe des combinaisons

$$(\alpha, \alpha_2, \ldots, \alpha_n), (\beta, \alpha_2, \ldots, \alpha_n), (\alpha, \gamma_2, \ldots, \gamma_n),$$

sur lesquelles f prend trois valeurs distinctes. Ces combinaisons engendrent un cube d'arête 2; c'est-à-dire un cube qui se projette sur chaque axe en deux points aux plus et dont la dimension est $\leqslant n$. Ce cube se définit comme

$$\{\alpha, \beta\} \times \{\alpha_2, \gamma_2\} \times \ldots \times \{\alpha_n, \gamma_n\}.$$

Considérons les hyperplans $x_1 = \alpha$ et $x_1 = \beta$ (voir fig. 2) de ce cube. Dans l'hyperplan $x_1 = \alpha$ relions le point $(\alpha_2, \ldots, \alpha_n)$ au point $(\gamma_2, \ldots, \gamma_n)$ par une suite d'arêtes contenues dans cet hyperplan (chaque arête relie un couple de combinaisons voisines, c'est-à-dire des combinaisons **appartenant** à l'hyperplan et différant en une coordonnée et une seule). Cette suite d'arêtes en définit une autre par projection sur l'hyperplan $x_1 = \beta$. Tout couple d'arêtes homologues forme un carré, si bien qu'on obtient une suite de carrés. Sur l'arête R_1 du premier carré, la fonction f prend les valeurs $f(\alpha, \alpha_2, \ldots, \alpha_n)$ et $f(\beta, \alpha_2, \ldots, \alpha_n)$; sur l'arête R_t du dernier carré, elle ne prend pas l'une au moins de ces valeurs. Dans ce cas, il existe dans cette suite un carré (de numéro i pour fixer les idées) tel que sur l'arête R_i la fonction prend les valeurs $f(\alpha, \alpha_2, \ldots, \alpha_n)$ et $f(\beta, \alpha_2, \ldots, \alpha_n)$ et sur l'arête R_{i+1} ne prend pas l'une au moins de ces valeurs. Le carré d'indice i est le carré cherché. Ce qui prouve le lemme.

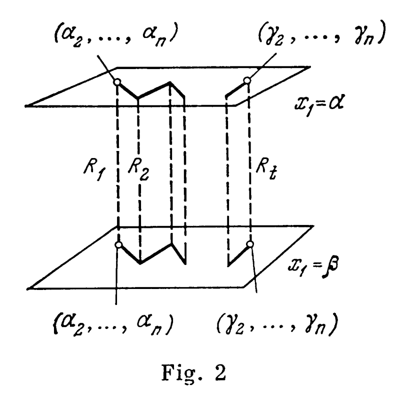


Fig. 2

Les lemmes prouvés admettent une interprétation géométrique simple. Soit $f(x_1, \ldots, x_n)$ une fonction essentielle prenant l valeurs, où $l \geqslant 3$. Alors:

1. Il existe un cube d'arête 2 sur lequel la fonction f prend au moins trois valeurs (lemme 3).
2. Il existe un cube d'arête $l - 1$ sur lequel la fonction f prend toutes les l valeurs (lemme 4).
3. Il existe un carré sur lequel la fonction f prend soit plus de deux valeurs, soit exactement deux valeurs, dont l'une en un seul point (lemme 5).

Remarque 1. Les lemmes 3 et 4 n'ont pas de sens pour $k = 2$, le lemme 5 a un sens, mais est faux, puisque la fonction

$$f(x_1, x_2) = x_1 + x_2 \pmod{2}$$

prend sur son domaine de définition, qui est un carré, deux valeurs, et toutes deux avec la multiplicité 2.

Remarque 2. Les lemmes 4 et 5 ne peuvent pas être renforcés. En effet, supposons que $3 \leqslant l \leqslant k - 1$, $n \geqslant 3$ et que

$$f_l(x_1, \ldots, x_n) = \begin{cases} i \text{ pour } x_1 = \ldots = x_n = i, \ i \leqslant l-1, \\ 0 \text{ en d'autres points.} \end{cases}$$

Soient $G_1, \ldots, G_n$ des ensembles quelconques de E^k et

$$1 \leqslant |G_1|, \ldots, |G_n| \leqslant l - 2.$$

Alors la fonction f_l ne peut prendre toutes les l valeurs sur $G_1 \times \ldots \times G_n$. D'autre part, f_l ne prend pas plus de deux valeurs sur tout carré.

Lorsqu'on étudie des fonctions $f(x_1, \ldots, x_n)$ de P_k douées de certaines propriétés sur un ensemble $\mathscr{E} = G_1 \times \ldots \times G_n$, on est souvent amené à passer à des fonctions $f'(x_1, \ldots, x_n)$ jouissant des mêmes propriétés, mais éventuellement sur un autre ensemble $\mathscr{E}' = G_1' \times \ldots \times G_n'$. Ce passage de la fonction f à la fonction f' sera appelé *normalisation*. La normalisation est donc reliée à une transformation des variables et des valeurs de la fonction f, cette transformation étant de la forme

$$f'(x_1, \ldots, x_n) = \psi(f(\psi_1(x_1), \ldots, \psi_n(x_n))).$$

Dans ces transformations, nous partons du fait que $|G_1'| = |G_1| = l_1, \ldots, |G_n'| = |G_n| = l_n$. Désignons par $\eta_0, \ldots, \eta_{l-1}$ les valeurs prises par f sur l'ensemble $\mathscr{E}$ et par $\eta_0', \ldots, \eta_{l-1}'$ une combinaison de nombres deux à deux distincts de E^k. La normalisation est donc définie par l'exhibition de correspondances biunivoques entre les ensembles:

$$\{\eta_0, \ldots, \eta_{l-1}\} \leftrightarrow \{\eta_0', \ldots, \eta_{l-1}'\},$$
$$G_1 \leftrightarrow G_1',$$
$$\cdot\ \cdot\ \cdot\ \cdot\ \cdot$$
$$G_n \leftrightarrow G_n'.$$

Ces correspondances biunivoques peuvent être assorties d'une condition subsidiaire consistant par exemple en ce qu'un point fixe $(\alpha_1, \ldots, \alpha_n)$ de $\mathscr{E}$ soit associé à un point fixe $(\alpha_1', \ldots, \alpha_n')$ de $\mathscr{E}'$ et η_i à η_j'. Il est évident que ces correspondances peuvent être réalisées par des fonctions $\psi(x), \psi_1(x), \ldots, \psi_n(x)$ de P_k que l'on peut toujours choisir dans l'ensemble des permutations; si $l, l_1, \ldots, l_n \leqslant k - 1$, on peut les choisir dans un ensemble de fonctions d'une variable, prenant au plus $k - 1$ valeurs. Il est évident que la normalisation laisse invariantes les multiplicités des valeurs de la fonction f sur $\mathscr{E}$. Dans la suite, on utilisera la normalisation sous les formes suivantes:

1) $\{\eta_0, \ldots, \eta_{l-1}\} = \{\eta_0', \ldots, \eta_{l-1}'\}$, $\psi(x) = x$,
$G_i' = \{0, \ldots, l_i - 1\}$ $(i = 1, \ldots, n)$.

2) $G_i = G_i'$, $\psi_i(x) = x$ $(i = 1, \ldots, n)$,
$\{\eta_0', \ldots, \eta_{l-1}'\} = \{0, \ldots, l - 1\}$.

3) $\{\eta_0', \ldots, \eta_{l-1}'\} = \{0, \ldots, l - 1\}$, $G_i' = \{0, \ldots, l_i - 1\}$
$(i = 1, \ldots, n)$.

Le cas 1 (changement des variables) et le cas 2 (changement des valeurs) sont des cas de normalisation incomplète. Pour points fixes η'_j et $(\alpha'_1, \ldots, \alpha'_n)$ on prend généralement 0 et $(0, \ldots, 0)$.

Prouvons maintenant un théorème qui généralise le théorème de Słupecki ([25]).

Théorème 7. *Supposons qu'un système* $\mathfrak{P}$ *de fonctions de* P_k, $k \geqslant 3$, *contient toutes les fonctions d'une variable, prenant au plus* $k - 1$ *valeurs. Pour que le système* $\mathfrak{P}$ *soit complet il est nécessaire et suffisant qu'il contienne une fonction essentielle* $f(x_1, \ldots, x_n)$ *qui prend toutes les* k *valeurs.*

Démonstration. *Nécessité.* Supposons que $\mathfrak{P}$ est un système complet et qu'il ne contient pas de fonction essentielle prenant toutes les k valeurs. Il est évident alors qu'on ne peut obtenir à partir de $\mathfrak{P}$ une fonction essentielle prenant toutes les k valeurs. Cette contradiction montre que $\mathfrak{P}$ contient une fonction essentielle prenant toutes les k valeurs.

Suffisance. Supposons que $\mathfrak{P}$ vérifie l'hypothèse du théorème et contient une fonction essentielle $f(x_1, \ldots, x_n)$ prenant toutes les k valeurs. Prouvons par récurrence que $\mathfrak{P}$ est complet.

1) *Base de la récurrence.* Montrons qu'à partir de $\mathfrak{P}$ on peut obtenir toutes les fonctions de P_k prenant deux valeurs. D'après le lemme 5 il existe un carré

$$(\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \alpha_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \beta_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \alpha_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \beta_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta_j, \alpha_{j+1}, \ldots, \alpha_n),$$
$$(\alpha_1, \ldots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta_j, \alpha_{j+1}, \ldots, \alpha_n),$$

où $\alpha_i \neq \beta_i$ et $\alpha_j \neq \beta_j$, sur lequel la fonction f prend au moins deux valeurs, dont l'une, η, en un seul point. Soit la fonction $\varphi(x)$:

$$\varphi(x) = \begin{cases} 0 \text{ pour } x = \eta, \\ 1 \text{ pour } x \neq \eta. \end{cases}$$

Il est évident que $\varphi(x) \in \mathfrak{P}$. Soit

$$g(x_1, x_2) = \varphi(f(\alpha_1, \ldots, \alpha_{i-1}, x_1, \alpha_{i+1}, \ldots, \alpha_{j-1}, x_2, \alpha_{j+1}, \ldots, \alpha_n)).$$

La fonction $g(x_1, x_2)$ prend sur le carré $\{(\alpha_i, \alpha_j), (\beta_i, \alpha_j), (\beta_i, \beta_j), (\alpha_i, \beta_j)\}$ deux valeurs, la valeur 0 en un point que nous désignerons par (α_1^0, α_2^0) et la valeur 1. En effectuant une normalisation incomplète telle que le point $(0, 0)$ se transforme en (α_1^0, α_2^0), on obtient une fonction $g'(x_1, x_2)$ telle que

$$g'(x_1, x_2) = g(\psi_1(x_1), \psi_2(x_2)),$$

où $\psi_1, \psi_2 \in \mathfrak{P}$. La fonction $g'(x_1, x_2)$ est visiblement un maximum sur l'ensemble $\{0, 1\} \times \{0, 1\}$. Désignons-la par $x_1 \vee_{01} x_2$. Comme le système $\mathfrak{P}$ contient les fonctions $j_i(x)$, où

$$j_i(x) = \begin{cases} 1 \text{ pour } x = i, \\ 0 \text{ pour } x \neq i, \end{cases}$$

il vient que

$$x_1 \&_{01} x_2 = j_0(j_0(x_1) \vee_{01} j_0(x_2))$$

est un minimum sur l'ensemble $\{0, 1\} \times \{0, 1\}$. Soit $h(x_1, \ldots, x_m)$, $h \not\equiv$ const, une fonction quelconque prenant les valeurs 0 et 1; alors

$$h(x_1, \ldots, x_m) = \bigvee_{(\sigma_1, \ldots, \sigma_m)} j_{\sigma_1}(x_1) \& \ldots \& j_{\sigma_m}(x_m) \& h(\sigma_1, \ldots, \sigma_m) =$$
$$= \bigvee_{01 \, (\sigma_1, \ldots, \sigma_m)} j_{\sigma_1}(x_1) \&_{01} \ldots \&_{01} j_{\sigma_m}(x_m) \&_{01} h(\sigma_1, \ldots, \sigma_m).$$

On peut donc obtenir la fonction $h(x_1, \ldots, x_m)$ à partir du système $\mathfrak{P}$. Comme $\mathfrak{P}$ contient toutes les fonctions d'une variable, qui prennent deux valeurs quelconques, on peut à partir de $\mathfrak{P}$ obtenir toutes les fonctions prenant deux valeurs quelconques.

2) Supposons qu'à partir de $\mathfrak{P}$ on ait construit toutes les fonctions prenant au plus $l - 1$ valeurs, $l - 1 < k$. Montrons qu'on peut alors construire toutes les fonctions de P_k prenant l valeurs.

Soit une fonction $f(x_1, \ldots, x_n)$. D'après le lemme 4 il existe n sous-ensembles $G_1, \ldots, G_n$, tels que $|G_i| \leqslant l - 1$ $(i = 1, \ldots, n)$ et que la fonction f prenne l valeurs $\eta_0, \eta_1, \ldots, \eta_{l-1}$ sur $\mathscr{E} = G_1 \times \ldots \times G_n$. On conviendra que ces valeurs sont prises respectivement sur les combinaisons suivantes de $\mathscr{E}$:

$$\tilde{\alpha}^{(0)} = (\alpha_1^{(0)}, \alpha_2^{(0)}, \ldots, \alpha_n^{(0)}),$$
$$\tilde{\alpha}^{(1)} = (\alpha_1^{(1)}, \alpha_2^{(1)}, \ldots, \alpha_n^{(1)}),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\tilde{\alpha}^{(l-1)} = (\alpha_1^{(l-1)}, \alpha_2^{(l-1)}, \ldots, \alpha_n^{(l-1)}),$$

i.e.

$$f(\tilde{\alpha}^{(0)}) = \eta_0, \; f(\tilde{\alpha}^{(1)}) = \eta_1, \; \ldots, \; f(\tilde{\alpha}^{(l-1)}) = \eta_{l-1}.$$

Montrons qu'avec les fonctions de $\mathfrak{P}$ on peut construire une fonction $h(x_1, \ldots, x_m)$ prenant les valeurs $\eta_0, \eta_1, \ldots, \eta_{l-1}$.

En effet, la fonction $h(x_1, \ldots, x_m)$ peut être définie à l'aide du tableau 10 dans lequel $\sigma = (\sigma_1, \ldots, \sigma_m)$. Définissons les fonctions $\psi_j(x_1, \ldots, x_m)$ $(j = 1, \ldots, n)$ comme indiqué dans le tableau 11. Alors

$$h(x_1, \ldots, x_m) = f(\psi_1(x_1, \ldots, x_m), \ldots, \psi_n(x_1, \ldots, x_m)),$$

Tableau 10

$x_1 \dots x_m$	$h(x_1, \dots, x_m)$
$\sigma_1 \dots \sigma_m$	$\eta_{i(\sigma)}$

Tableau 11

$x_1 \dots x_m$	$\psi_j(x_1, \dots, x_m)$
$\sigma_1 \dots \sigma_m$	$\alpha_j^{(i(\sigma))}$

car

$$f(\psi_1(\sigma_1, \dots, \sigma_m), \dots, \psi_n(\sigma_1, \dots, \sigma_m)) = f(\alpha_1^{(i(\sigma))}, \dots, \alpha_n^{(i(\sigma))}) = \eta_{i(\sigma)},$$

$$h(\sigma_1, \dots, \sigma_m) = \eta_{i(\sigma)}.$$

Si l'on dispose de toutes les fonctions prenant les l valeurs données $\eta_0, \dots, \eta_{l-1}$, on peut, dans le cas $l < k$, obtenir à l'aide de fonctions d'une variable, qui prennent au moins k valeurs les fonctions restantes prenant l valeurs. Donc, en appliquant cette procédure on arrive jusqu'à $l = k$ et l'on construit ainsi toutes les fonctions de P_k. Ceci prouve la condition suffisante.

Corollaire (critère de Słupecki). *Supposons qu'un système $\mathfrak{P}$ de fonctions de P_k, $k \geqslant 3$, contient toutes les fonctions à une variable. Pour que ce système soit complet il est nécessaire et suffisant qu'il contienne une fonction essentielle $f(x_1, \dots, x_n)$ prenant toutes les k valeurs.*

Remarque. Le théorème prouvé est vrai pour $k \geqslant 3$ et ne peut être généralisé au cas $k = 2$. En effet, le système

$$\mathfrak{P} = \{0, 1, x, \bar{x}, x_1 + x_2\}$$

n'est pas complet, puisque $\mathfrak{P} \subseteq L$.

L'utilisation directe de ce théorème et à plus forte raison de ses corollaires n'est pas toujours commode, car il faut préalablement établir l'existence dans $\mathfrak{P}$ de toutes les fonctions d'une variable, prenant au plus $k - 1$ valeurs, c'est-à-dire $k^k - k!$ fonctions. Les difficultés soulevées par ces procédures croissent fortement avec $\mathfrak{P}$. On a donc intérêt à remplacer dans les énoncés des théorèmes la condition que le système contienne un ensemble donné de fonctions d'une variable par la condition qu'il engendre cet ensemble de fonctions d'une variable. Ceci peut être établi d'une manière bien plus économique : par exemple si l'on sait qu'à partir de $\mathfrak{P}$ on peut obtenir des systèmes concrets de fonctions d'une variable engendrant toutes les fonctions d'une variable.

Voici deux exemples de tels systèmes.

Théorème 8 (S. Piccard [18]). *Toutes les fonctions d'une variable de P_k peuvent être engendrées par trois fonctions:*

$$f(x) = x - 1 \ (\text{mod. } k),$$

$$g(x) = \begin{cases} x, & 0 \leqslant x \leqslant k-3, \\ k-1, & x = k-2, \\ k-2, & x = k-1, \end{cases} \qquad h(x) = \begin{cases} 1, & x = 0, \\ x, & x \neq 0. \end{cases}$$

Théorème 9. *Toutes les fonctions d'une variable de P_k peuvent être engendrées par k fonctions*

$$f_i(x) = \begin{cases} i \ \textit{pour} \ x = 0, \\ 0 \ \textit{pour} \ x = i, \quad (i = 1, \ldots, k-1) \\ x \ \textit{dans les autres cas} \end{cases}$$

et la fonction $h(x)$.

Nous glisserons sur les démonstrations de ces théorèmes, car elles sont simples.

Voyons en conclusion encore une application du critère de complétude. Nous allons donner une propriété caractéristique d'une fonction de P_k (fonction de Sheffer) qui forme un système complet. Cette propriété renforce légèrement le théorème de Martin.

Théorème 10. *Une fonction $f(x_1, \ldots, x_n)$ de P_k, où $k \geqslant 3$, est une fonction de Sheffer si et seulement si $f(x_1, \ldots, x_n)$ engendre toutes les fonctions d'une variable, prenant au plus $k - 1$ valeurs.*

Démonstration. La condition nécessaire est évidente.

Condition suffisante. Il est évident que $f(x_1, \ldots, x_n)$ doit prendre toutes les k valeurs, puisqu'à partir d'elle on peut par exemple obtenir toutes les constantes. Si f est une fonction inessentielle, alors c'est une permutation et elle ne peut donner que des permutations. Dans ce cas on ne peut même pas obtenir des constantes. Ce qui est impossible, donc f est une fonction essentielle. En utilisant le critère de complétude, on déduit que le système $\mathfrak{P} = \{f(x_1, \ldots, x_n)\}$ est complet. C.q.f.d.

§ 12. Particularités des logiques k-valentes

L'exposé précédent montre que les logiques k-valentes présentent beaucoup d'affinités avec la logique bivalente. De nombreux résultats de la logique bivalente sont valables pour les logiques k-valentes. Certes, la croissance de k apporte des complications notoires dans les formulations et les démonstrations.

Mais d'ores et déjà on a accumulé suffisamment de faits mettant en évidence l'originalité des logiques k-valentes, voire même des

faits soulignant la différence fondamentale existant entre P_k pour $k \geqslant 3$ et P_2. Ces faits méritent d'être retenus compte tenu du travail de Post [21] qui a avancé l'idée d'une réduction des logiques k-valentes à la logique bivalente. Plus exactement, Post a proposé de considérer à la place de la fonction $f(x_1, \ldots, x_n)$ de P_k un système de fonctions

$$\varphi_1 (x_{11}, \ldots, x_{1l}, \ldots, x_{n1}, \ldots, x_{nl}), \ldots$$
$$\ldots, \varphi_l (x_{11}, \ldots, x_{1l}, \ldots, x_{n1}, \ldots, x_{nl}),$$

où $l =]\log_2 k[$ *), de plus si la valeur α_i admet la représentation binaire $\alpha_{i1} \ldots \alpha_{il}$ $(i = 1, \ldots, n)$, la valeur $f(\alpha_1, \ldots, \alpha_n)$ admettra la représentation binaire

$$\varphi_1 (\alpha_{11}, \ldots, \alpha_{1l}, \ldots, \alpha_{n1}, \ldots, \alpha_{nl}) \ldots$$
$$\ldots \varphi_l (\alpha_{11}, \ldots, \alpha_{1l}, \ldots, \alpha_{n1}, \ldots, \alpha_{nl}).$$

Ceci étant, à l'opération de superposition dans P_k correspondra une opération spéciale sur les systèmes de fonctions $\{\varphi_1, \ldots, \varphi_l\}$ de P_2. La possibilité de coder les fonctions de P_k avec des systèmes de fonctions de P_2 peut être utile lors de la résolution de problèmes logiques sur calculateurs, mais présente peu d'intérêt dans les logiques k-valentes. La spécificité de P_k pour $k \geqslant 3$ a été soulignée déjà dans les travaux de Słupecki [25], ainsi que dans ceux de Kouznetsov [5] et de Yablonski [28, 29]. Cependant, des résultats ultérieurs ont montré que la distinction entre P_k et P_2 était bien plus profonde qu'elle ne le paraissait auparavant.

Dans ce paragraphe on se propose de toucher à quelques aspects seulement de ce problème. On s'intéressera notamment aux trois problèmes suivants.

1. Existence de bases pour les classes fermées de P_k.
2. Puissance du système de toutes les classes fermées de P_k.
3. Possibilité de représenter les fonctions de P_k par des polynômes.

Dans P_2, les théorèmes de Post et de Gégalkine nous permettent de donner les réponses suivantes à ces problèmes.

1. Toute classe fermée de P_2 possède une base finie.
2. La puissance de l'ensemble de toutes les classes fermées de P_2 est égale à $\aleph_0$.
3. Toute fonction de P_2 peut être représentée par un polynôme mod. 2.

Pour P_k $(k \geqslant 3)$ la réponse à ces problèmes est donnée par les théorèmes suivants.

Théorème 11 (Yanov [32]). *Pour tout k $(k \geqslant 3)$ il existe dans P_k une classe fermée ne possédant pas de base.*

---

*) $]a[$ désigne le plus petit entier non inférieur à a.

Démonstration. Considérons la suite de fonctions

$$f_0 = 0, \ f_i(x_1, \ldots, x_i) = \begin{cases} 1 \text{ pour } x_1 = \ldots = x_i = 2, \\ 0 \text{ dans les autres cas} \end{cases} \quad (i = 1, 2, \ldots).$$

Désignons par $\mathfrak{M}_k$ l'ensemble des fonctions obtenues à partir de $\{f_0, f_1, \ldots\}$ en changeant le nom des variables (sans les identifier). Il est immédiat de voir que $\mathfrak{M}_k$ est une classe fermée. Supposons que $\mathfrak{M}_k$ possède une base. Alors cette base contient une fonction $\tilde{f}$ obtenue à partir de la fonction f_{n_0} en changeant le nom des variables, pour laquelle n_0 est minimal. Deux cas sont possibles.

1. La base contient encore au moins une fonction $\tilde{f}'$. A cette fonction correspond une fonction f_{n_1}, $n_1 > n_0$. Comme f_{n_0} peut être obtenue à partir de f_{n_1} par identification des variables, $\tilde{f}$ s'exprime en fonction de $\tilde{f}'$, ce qui contredit la définition de la base.

2. La base est composée d'une seule fonction f. Dans ce cas, aucune fonction f_n pour $n > n_0$ ne peut être obtenue à partir de f, puisque $f_{n_0}(\ldots, f_{n_0}, \ldots) \equiv 0$. Nous aboutissons de nouveau à une contradiction.

Il reste donc à admettre que $\mathfrak{M}_k$ ne possède pas de base. Ceci achève la démonstration du théorème.

Théorème 12 (Moutchnik [32]). *Pour tout k ($k \geqslant 3$) il existe dans P_k une classe fermée munie d'une base dénombrable.*

Démonstration. Considérons la suite de fonctions ($i = 2, 3, \ldots$)

$$f_i(x_1, \ldots, x_i) = \begin{cases} 1 \text{ pour } x_1 = \ldots = x_{j-1} = x_{j+1} = \ldots = x_i = 2, \quad x_j = 1 \\ \qquad\qquad\qquad\qquad (j = 1, \ldots, i), \\ 0 \text{ dans les autres cas.} \end{cases}$$

Désignons par $\mathfrak{M}_k$ la classe fermée engendrée par le système $\{f_2, f_3, \ldots\}$. Montrons que ce système est une base de $\mathfrak{M}_k$. Pour cela il suffit d'établir qu'aucune des fonctions f_m ne peut être exprimé par les autres fonctions du système, c'est-à-dire que la représentation

$$f_m = \mathfrak{A}[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots]$$

est impossible.

En développant la formule $\mathfrak{A}$ on obtient

$$\mathfrak{A}[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots] =$$
$$= f_r(\mathfrak{B}_1[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots], \ldots, \mathfrak{B}_r[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots])$$

ou

$$f_m(x_1, \ldots, x_m) =$$
$$= f_r(\mathfrak{B}_1[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots], \ldots, \mathfrak{B}_r[f_2, \ldots, f_{m-1}, f_{m+1}, \ldots]).$$

On distinguera trois cas à priori.

1. Deux au moins des formules $\mathfrak{B}_1, \ldots, \mathfrak{B}_r$ $(r \geqslant 2)$ ne sont pas des symboles de variables. Alors pour toutes valeurs des variables $x_1, \ldots, x_m$, on ne peut avoir que les valeurs 0 et 1 à la place des formules correspondantes dans la fonction f_r. Donc, le second membre sera identiquement nul, ce qui contredit la possibilité de cette représentation, puique $f_m \not\equiv 0$.

2. Parmi les formules $\mathfrak{B}_1, \ldots, \mathfrak{B}_r$, seule une, $\mathfrak{B}_s$, n'est pas un symbole de variable. Les autres formules se réduisant par hypothèse à des variables et puisque $r \geqslant 2$, il existe au moins une formule $\mathfrak{B}_p \equiv x_q$. Considérons la combinaison $x_1 = \ldots = x_{q-1} = x_{q+1} = \ldots = x_m = 2$ et $x_q = 1$. Sur cette combinaison la formule $\mathfrak{B}_s$ prend soit la valeur 0, soit la valeur 1. Donc, pour ces valeurs des variables, en deux endroits de la fonction f_r on aura des valeurs différentes de 2. Donc, le second membre prend la valeur 0. Or, le premier membre prend la valeur 1 sur cette combinaison. Nous avons abouti à une contradiction.

3. Toutes les formules $\mathfrak{B}_1, \ldots, \mathfrak{B}_r$ sont des symboles de variables. Dans ce cas $r > m$ et par suite dans la formule on trouve au moins deux entrées d'une variable x_p. Considérons la combinaison $x_1 = \ldots = x_{p-1} = x_{p+1} = \ldots = x_m = 2$ et $x_p = 1$. On constate que sur cette combinaison le premier membre prend la valeur 1 et le second, la valeur 0. Donc, ce cas est aussi impossible. C.q.f.d.

Au théorème prouvé se rattache le

Théorème 13. *Pour tout k $(k \geqslant 3)$ l'ensemble de classes fermées contenues dans P_k a la puissance du continu.*

Démonstration. Le nombre de classes fermées de P_k peut être majoré par le nombre de tous les sous-ensembles de fonctions de P_k. Comme P_k est un ensemble de fonctions dénombrable, le nombre de sous-ensembles de P_k est égal au continu.

Pour achever la démonstration il faut minorer le nombre de classes fermées de P_k. Considérons à cet effet la classe fermée $\mathfrak{M}_k$ construite dans la démonstration du théorème précédent. Cette classe possède la base

$$\{f_2, f_3, \ldots\}.$$

Pour chaque suite $\{\rho_1, \rho_2, \ldots\}$, où $2 \leqslant \rho_1 < \rho_2 < \ldots$ considérons la classe $\mathfrak{M}_k(\rho_1, \rho_2, \ldots)$ engendrée par le système de fonctions $\{f_{\rho_1}, f_{\rho_2}, \ldots\}$. Il est immédiat de voir que

$$\mathfrak{M}_k(\rho_1', \rho_2', \ldots) \neq \mathfrak{M}_k(\rho_1'', \rho_2'', \ldots),$$

si

$$\{\rho_1', \rho_2', \ldots\} \neq \{\rho_1'', \rho_2'', \ldots\}.$$

Donc, la famille $\{\mathfrak{M}_k(\rho_1, \rho_2, \ldots)\}$ a la puissance du continu. C.q.f.d.

Théorème 14. *Le système des polynômes* mod. k *est complet dans* P_k *si et seulement si* $k = p$, *où* p *est un nombre premier.*

Démonstration. Il est aisé de voir que pour toute fonction $f(x_1, \ldots, x_n)$ de P_k on a la représentation

$$f(x_1, \ldots, x_n) = \sum_{(\sigma_1, \ldots, \sigma_n)} j_{\sigma_1}(x_1) \ldots j_{\sigma_n}(x_n) f(\sigma_1, \ldots, \sigma_n) \pmod{k}.$$

Donc, le problème de la représentation de la fonction f par des polynômes mod. k se réduit à celui de la représentation par des polynômes des fonctions

$$j_0(x), \ldots, j_{k-1}(x).$$

Puisque

$$j_\sigma(x) = j_0(x - \sigma),$$

on peut affirmer que le système des polynômes mod. k est complet si et seulement si la fonction $j_0(x)$ se représente par un polynôme.

1. Supposons que $k = p$. En se servant du premier théorème de Fermat on trouve

$$j_0(x) = 1 - x^{p-1} \pmod{p},$$

c'est-à-dire que le système des polynômes est complet dans P_k *).

Voici une autre méthode de résolution de ce problème. Cherchons la représentation de la fonction d'une variable $g(x)$ sous forme d'un polynôme par la méthode des coefficients indéterminés:

$$g(x) = a_0 + a_1 x + \ldots + a_{p-1} x^{p-1}.$$

On obtient le système d'équations

$$a_0 + a_1 \cdot 0^1 + a_2 \cdot 0^2 + \ldots + a_{p-1} \cdot 0^{p-1} = g(0),$$
$$a_0 + a_1 \cdot 1^1 + a_2 \cdot 1^2 + \ldots + a_{p-1} \cdot 1^{p-1} = g(1),$$
$$a_0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + \ldots + a_{p-1} \cdot 2^{p-1} = g(2),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$a_0 + a_1 (p-1)^1 + a_2 (p-1)^2 + \ldots + a_{p-1} (p-1)^{p-1} = g(p-1)$$

---

*) Voici une démonstration du premier théorème de Fermat. Supposons que $1 \leqslant a \leqslant p - 1$. Alors les nombres $r_1 = a \cdot 1, \ldots, r_{p-1} = a(p-1)$ ne sont pas congrus mod. p. Donc, $r_1 \ldots r_{p-1} \equiv a^{p-1}(p-1)! \pmod{p}$ et $(p-1)! \equiv a^{p-1}(p-1)! \pmod{p}$ ou $1 \equiv a^{p-1} \pmod{p}$.

dont le déterminant

$$\Delta = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1^2 & \dots & 1^{p-1} \\ 1 & 2 & 2^2 & \dots & 2^{p-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & p-1 & (p-1)^2 & \dots & (p-1)^{p-1} \end{vmatrix}$$

est un déterminant de Vandermonde. On sait que

$$\Delta = \prod_{1 \leqslant i < j \leqslant p-1} (j-i).$$

Comme p est premier, $\Delta \not\equiv 0$ (mod. p). En se servant de la règle de Cramer et puisque $\Delta \not\equiv 0$ (mod. p), on peut résoudre en nombres entiers les congruences

$$a_i \Delta \equiv \Delta_i \text{ (mod. } p) \quad (i = 0, \dots, p-1),$$

où Δ_i est le mineur correspondant. On obtient ainsi l'unique solution du système initial et par conséquent le polynôme représentant $g(x)$.

2. Supposons que $k \neq p$. Alors $k = k_1 k_2$, où $k > k_2 > 1$. Supposons que

$$j_0(x) = b_0 + b_1 x + \dots + b_s x^s \text{ (mod. } k).$$

Pour $x = 0$ on obtient $b_0 = 1$. Pour $x = k_1$, on a

$$0 = 1 + b_1 k_1 + \dots + b_s k_1^s \text{ (mod. } k)$$

ou

$$k - 1 = b_1 k_1 + \dots + b_s k_1^s \text{ (mod. } k),$$

c'est-à-dire que $k - 1$ se divise par k_1. Donc, k et $k - 1$ se divisent aussi par k_1, ce qui n'est possible que pour $k_1 = 1$. Nous avons abouti à une contradiction. Donc, pour $k \neq p$ la fonction $j_0(x)$ n'est pas représentable par un polynôme mod. k.

Le théorème prouvé se généralise aisément au cas où E^k peut être muni de deux opérations : l'addition $\oplus$ et la multiplication $\times$ pour lesquelles E^k est un corps commutatif. On montre en algèbre qu'une condition nécessaire et suffisante d'existence d'un corps commutatif fini ou corps de Galois est que $k = p^m$. Ce corps est défini de façon unique à un isomorphisme près. Il forme en outre un groupe abélien de caractéristique p pour l'addition, c'est-à-dire que pour tout élément α on a

$$\underbrace{\alpha \oplus \dots \oplus \alpha}_{p} = 0,$$

où 0 est l'élément neutre du groupe. On définit ce groupe en traitant les nombres α comme des nombres dans un système de numération

de base p, c'est-à-dire sous forme de combinaisons $(\alpha_1, \ldots, \alpha_m)$ et en considérant l'opération $\alpha \oplus \beta = (\alpha_1 + \beta_1, \ldots, \alpha_m + \beta_m)$ (le symbole $+$ représente l'addition modulo p). Tous les éléments du corps de Galois à l'exception de 0 forment un groupe cyclique pour la deuxième opération.

Exemple. Soit $k = 2^2$. Alors l'opération $\oplus$ est de la forme représentée sur le tableau 12. Pour construire le tableau de l'opération $\times$

Tableau 12

$\oplus$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Tableau 13

$\times$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

on remarquera que les nombres 1, 2 et 3 peuvent être exprimés comme les puissances d'un élément α (ceci résulte de ce qu'un groupe multiplicatif est cyclique). Le nombre α vérifie l'équation

$$\alpha^3 = 1$$

ou, puisque $\alpha \neq 1$,

$$\alpha^2 \oplus \alpha \oplus 1 = 0.$$

En prenant l'élément 2 pour α on obtient $\alpha^2 = \ominus (\alpha \oplus 1) = 3$. Ceci nous donne le tableau de $\times$ (tableau 13), puisque

$$2 \cdot 2 = \alpha \cdot \alpha = 3,$$
$$2 \cdot 3 = \alpha \cdot \alpha^2 = \alpha^3 = 1,$$
$$3 \cdot 3 = \alpha^2 \cdot \alpha^2 = \alpha^4 = \alpha = 2.$$

On peut, en répétant la première partie de la démonstration du théorème précédent, montrer que toute fonction $f(x_1, \ldots, x_n)$ de P_k, pour $k = p^m$, est représentable par un polynôme sur le corps de Galois correspondant.

En particulier, toute fonction de P_4 se représente par des polynômes sur le corps de Galois, mais pas par des polynômes modulo 4. Ainsi donc, pour P_k $(k \geqslant 3)$ on obtient les réponses suivantes aux trois questions posées.

1. Dans P_k il existe des classes fermées ne possédant pas de base finie.

2. La puissance de l'ensemble de toutes les classes fermées de P_k est égale à c.

3. Toute fonction de P_k peut être représentée par un polynôme modulo k (resp., sur le corps de Galois) si et seulement si $k = p$ (resp., lorsque $k = p^m$).

On voit donc, en comparant ces réponses avec celles qui correspondent à $k = 2$, qu'il existe une différence fondamentale entre la logique bivalente et les logiques indiquées, et de plus que certains problèmes se résolvent de manière différente selon la valeur de k.

CHAPITRE 3

FONCTIONS DÉTERMINÉES BORNÉES (AUTOMATES) ET OPÉRATIONS SUR CES FONCTIONS

Nous avons fait connaissance de deux systèmes fonctionnels munis d'opérations:

l'algèbre logique (P_2, S), système de fonctions booléennes muni de l'opération de superposition;

la logique k-valente (P_k, S), système de fonctions de P_k muni de l'opération de superposition.

En faisant varier l'objet fonctionnel et l'opération on obtiendrait d'autres systèmes. Ainsi, en compliquant les objets fonctionnels, on obtient de façon naturelle:

la logique $\aleph_0$-valente $(P_{\aleph_0}, S)$, système muni de l'opération de superposition et contenant les constantes $0, 1, \ldots, k, \ldots$ et des fonctions $f(x_1, \ldots, x_n)$ à valeurs dans $E^{\aleph_0}$ et dont les variables sont définies sur la série numérique achevée $E^{\aleph_0} = \{0, 1, 2, \ldots\}$;

la logique $\mathfrak{c}$-valente $(P_{\mathfrak{c}}, S)$, système muni de l'opération de superposition et contenant les constantes de $[0, 1]$ et des fonctions à valeurs dans $[0, 1]$ et dont les variables sont définies sur $[0, 1]$.

Nous nous intéresserons à d'autres systèmes plus importants. Dans ce chapitre il sera question d'un système fonctionnel rattaché aux automates.

§ 13. Fonctions déterminées

L'être fonctionnel que nous allons étudier est une variété de la logique $\mathfrak{c}$-valente. Au lieu des nombres réels de l'intervalle $[0, 1]$ nous prendrons l'ensemble $E_k^{\mathfrak{c}}$ de toutes les séquences k-valentes α, où

$$\alpha = \{\alpha(1), \alpha(2), \ldots, \alpha(m), \ldots\},$$
$$\alpha(i) \in E^k \text{ pour tous les } i \ (i = 1, 2, \ldots).$$

Désignons par $P_{\mathfrak{c}}^k$ l'ensemble de toutes les fonctions

$$f(x_1, \ldots, x_n),$$

définies sur les combinaisons $(\alpha_1, \ldots, \alpha_n)$, où $\alpha_i \in E_k^{\mathfrak{c}}$ ($i = 1, 2, \ldots, n$), et à valeurs dans $E_k^{\mathfrak{c}}$. Donc, les fonctions de $P_{\mathfrak{c}}^k$ transforment les combinaisons de séquences k-valentes en séquences k-valentes. Incluons dans $P_{\mathfrak{c}}^k$ toutes les séquences de $E_k^{\mathfrak{c}}$ en les considérant comme des fonctions dépendant d'un ensemble vide de variables ($n = 0$), c'est-à-dire comme des constantes.

Exemple 1. Supposons que $k = 2$ et que

$$f(\alpha) = \begin{cases} (0,\ 0,\ \ldots) & \text{si } \alpha = (0,\ 0,\ \ldots), \\ (1,\ 1,\ \ldots) & \text{si } \alpha \neq (0,\ 0,\ \ldots). \end{cases}$$

Il est évident que

$$f(x) \in P_{\mathfrak{c}}^2.$$

On remarquera que les fonctions de $P_{\mathfrak{c}}^k$ ne peuvent pas être définies par un tableau, car l'ensemble $E_k^{\mathfrak{c}}$ (et partant, l'ensemble des « lignes » du tableau) a la puissance du continu. Il s'ensuit que l'ensemble des fonctions de $P_{\mathfrak{c}}^k$ dépendant des variables $x_1, x_2, \ldots$ a la puissance de l'hypercontinu. Cette situation nous contraint à étudier dans la suite un être fonctionnel plus étroit.

On se servira de l'écriture vectorielle pour représenter dans la suite les combinaisons et fonctions. Ainsi, on écrira X pour $(x_1, \ldots, x_n)$ et $f(X)$ pour $f(x_1, \ldots, x_n)$. Ceci étant, la valeur de la variable X est un vecteur (une combinaison) $\alpha = (\alpha_1, \ldots, \alpha_n)$ dont les composantes sont des séquences k-valentes:

$$\alpha_i = \{\alpha_i(1),\ \alpha_i(2),\ \ldots,\ \alpha_i(m),\ \ldots\} \quad (i = 1,\ 2,\ \ldots,\ n).$$

On traitera α comme une séquence de vecteurs

$$\alpha = \{\alpha(1),\ \alpha(2),\ \ldots,\ \alpha(m),\ \ldots\},$$

où

$$\alpha(i) = (\alpha_1(i),\ \alpha_2(i),\ \ldots,\ \alpha_n(i)) \quad (i = 1,\ 2,\ \ldots).$$

On admet donc qu'est réalisée l'identité

$$\begin{gathered}(\{\alpha_1(1),\ \alpha_1(2),\ \ldots,\ \alpha_1(m),\ \ldots\}, \\ \{\alpha_2(1),\ \alpha_2(2),\ \ldots,\ \alpha_2(m),\ \ldots,\},\ \ldots \\ \ldots,\ \{\alpha_n(1),\ \alpha_n(2),\ \ldots,\ \alpha_n(m),\ \ldots\}) \equiv \\ \equiv \{(\alpha_1(1),\ \alpha_2(1),\ \ldots,\ \alpha_n(1)),\ (\alpha_1(2),\ \alpha_2(2),\ \ldots,\ \alpha_n(2)),\ \ldots \\ \ldots,\ (\alpha_1(m),\ \alpha_2(m),\ \ldots,\ \alpha_n(m)),\ \ldots\}.\end{gathered}$$

Cette séquence de vecteurs peut être considérée comme une séquence de combinaisons $(\alpha_1(i), \alpha_2(i), \ldots, \alpha_n(i))$ ou de nombres dans un système de numération à base k. Chacun de ces nombres appartient à l'ensemble E^N, où $N = k^n$.

Donc, toute fonction $f(x_1, \ldots, x_n)$ de $P_{\mathfrak{C}}^k$ peut être considérée comme une fonction $f(X)$ de l'ensemble $P_{\mathfrak{C}}^N$, mais dépendant d'une seule variable (et à valeurs dans $E^k \subset E^N$). On pourra ainsi ramener l'étude d'une fonction $f(x_1, \ldots, x_n)$ de $P_{\mathfrak{C}}^k$ à celle d'une fonction $f(X)$ d'une variable de $P_{\mathfrak{C}}^N$, où $N = k^n$. Cette réduction repose sur des raisonnements formels liés à l'interprétation d'une combinaison de séquences comme une séquence de combinaisons. Nous verrons plus bas que pour une certaine classe de fonctions de $P_{\mathfrak{C}}^k$ cette interprétation admet une signification physique.

Définition. On dit qu'une fonction $f(X)$ de $P_{\mathfrak{C}}^N$ est *déterminée* si pour tout m et pour toutes séquences α et β telles que

$$\alpha(1) = \beta(1),\ \alpha(2) = \beta(2),\ \ldots,\ \alpha(m) = \beta(m),$$

les valeurs γ et δ de la fonction f, où $\gamma = f(\alpha)$ et $\delta = f(\beta)$, sont des séquences dont les m premiers termes sont confondus, c'est-à-dire que

$$\gamma(1) = \delta(1),\ \gamma(2) = \delta(2),\ \ldots,\ \gamma(m) = \delta(m).$$

L'ensemble de toutes les fonctions déterminées, que nous désignerons par $P_{\mathfrak{d}}^k$, contient visiblement toutes les constantes de $P_{\mathfrak{C}}^k$.

Supposons que $f(\alpha) = \gamma$. De la définition il s'ensuit que la valeur $\gamma(m)$ du m-ième terme ($m = 1, 2, \ldots$) de la séquence γ est complètement déterminée par celles des m premiers termes

$$\alpha(1),\ \alpha(2),\ \ldots,\ \alpha(m)$$

de la séquence α, c'est-à-dire que

$$\gamma(m) = f_m(\alpha(1),\ \alpha(2),\ \ldots,\ \alpha(m)).$$

Comme

$$f_m(\alpha(1),\ \alpha(2),\ \ldots,\ \alpha(m)) = \\ = f_m(\alpha_1(1),\ \ldots,\ \alpha_n(1),\ \alpha_1(2),\ \ldots,\ \alpha_n(m)),$$

il est clair que f_m est une fonction de P_k dépendant de nm variables.

Donc, la fonction déterminée $f(X)$ est définie par une suite de fonctions de P_k

$$f \sim \{f_1, f_2, \ldots, f_m, \ldots\},$$

où

$$f_1 = f_1(X_1),$$
$$f_2 = f_2(X_1, X_2),$$
$$\cdots\cdots$$
$$f_m = f_m(X_1, X_2, \ldots, X_m),$$
$$\cdots\cdots\cdots\cdots$$

$$X_1 = (x_1 (1), x_2 (1), \ldots, x_n (1)),$$
$$X_2 = (x_1 (2), x_2 (2), \ldots, x_n (2)),$$
$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$
$$X_m = (x_1 (m), x_2 (m), \ldots, x_n (m)),$$
$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

La fonction déterminée $f (x_1, \ldots, x_n)$ peut être interprétée comme ceci. Supposons qu'on soit en possession d'un « convertis-

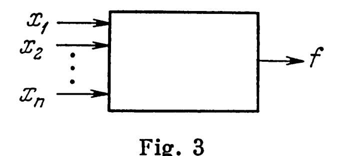


Fig. 3

seur discret » (fig. 3) à *n entrées* $x_1, \ldots, x_n$ et une *sortie* f. Aux *instants* $t = 1, 2, \ldots, m, \ldots$, on applique aux entrées les séquences (d'entrée)

$$\alpha_1 = \{\alpha_1 (1), \alpha_1 (2), \ldots, \alpha_1 (m), \ldots\},$$
$$\alpha_2 = \{\alpha_2 (1), \alpha_2 (2), \ldots, \alpha_2 (m), \ldots\},$$
$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$
$$\alpha_n = \{\alpha_n (1), \alpha_n (2), \ldots, \alpha_n (m), \ldots\}.$$

Aux mêmes instants t est délivrée en sortie la séquence (de sortie) $\gamma = \{\gamma (1), \gamma (2), \ldots, \gamma (m), \ldots\}$ et de plus $\gamma = f (\alpha_1, \alpha_2, \ldots, \alpha_n)$. Il est évident que dans un convertisseur discret la valeur $\gamma (m)$ ne dépend que des valeurs des séquences d'entrée aux instants $t = 1, 2, \ldots, m$ et pas des valeurs futures. Donc, la transformée f est une fonction déterminée.

A noter que les constantes de P_{d}^k $(n = 0)$ s'interprètent à l'aide d'un convertisseur discret sans entrées (un « générateur »). A noter encore que les séquences appliquées aux entrées du convertisseur, c'est-à-dire $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ peuvent être traitées comme une séquence de combinaisons

$$\{(\alpha_1 (1), \alpha_2 (1), \ldots, \alpha_n (1)), (\alpha_1 (2), \alpha_2 (2), \ldots, \alpha_n (2)), \ldots\}.$$

L'identité que nous avons introduite est réalisée ici de façon naturelle.

Le théorème qui va suivre résulte du fait qu'une fonction déterminée $f (x_1, \ldots, x_n)$ est complètement définie par une suite de fonctions de P_k.

Théorème 1. *L'ensemble de toutes les fonctions déterminées des variables* $x_1, x_2, \ldots, x_n$ *a la puissance du continu.*

Voyons quelques exemples en conclusion.

Exemple 2. a) La fonction $f_\Phi(x_1, \ldots, x_n)$, où $\Phi(x_1, \ldots, x_n) \in P_k$, est définie comme ceci:

$$f_\Phi(x_1, \ldots, x_n) \sim \{\Phi(x_1(1), \ldots, x_n(1)),$$
$$\Phi(x_1(2), \ldots, x_n(2)), \ldots, \Phi(x_1(m), \ldots, x_n(m)), \ldots\}.$$

Les valeurs de la fonction f_Φ s'obtiennent par le calcul de la fonction Φ sur les termes correspondants des séquences d'entrée. D'où $f_\Phi \in P^k_\text{d}$. En particulier, si l'on prend $\Phi(x_1, x_2) = x_1 \& x_2$ ($k = 2$), alors

$$f_\& (x_1, x_2) = \{x_1(1) \& x_2(1),\ x_1(2) \& x_2(2), \ldots, x_1(m) \& x_2(m), \ldots\}.$$

La séquence de sortie est ici la conjonction terme à terme des séquences d'entrée.

Désignons par P^k l'ensemble de toutes les fonctions f_Φ, où $\Phi \in P_k$.

b) La fonction $z = x + y$ qui est la somme de deux séquences k-valentes est donnée par un algorithme ordinaire d'addition rang par rang de deux nombres à une infinité de rangs dans un système de numération à base k:

$$\begin{array}{r} \ldots\ x(3),\ x(2),\ x(1) \\ +\ \ldots\ y(3),\ y(2),\ y(1) \\ \hline \ldots\ z(3),\ z(2),\ z(1) \end{array}$$

Il est clair que $z(m)$ est définie par les m premiers termes. Donc, $x + y \in P^k_\text{d}$.

c) La fonction x^2 est définie par un algorithme de multiplication de deux nombres à une infinité de rangs dans un système de numération à base k:

$$\begin{array}{rlll} & \ldots\ x(3), & x(2), & x(1) \\ \times & \ldots\ x(3), & x(2), & x(1) \\ \hline & \ldots\ x(3)\,x(1), & x(2)\,x(1), & x(1)\,x(1) \\ + & \ldots\ x(2)\,x(2), & x(1)\,x(2) & \\ & \ldots\ x(1)\,x(3) & & \\ & \ldots\ldots\ldots & \ldots\ldots\ldots & \ldots\ldots\ldots \\ \hline & \ldots\ z(3), & z(2), & z(1) \end{array}$$

$z(m)$ est complètement définie par les m premiers termes de la séquence d'entrée. Donc, $x^2 \in P^k_\text{d}$.

On peut facilement citer des exemples de fonctions de $P_{\mathfrak{c}}^{k}$ qui ne sont pas déterminées. Ainsi, la fonction $f(x)$ de l'exemple 1 de ce paragraphe n'est pas déterminée.

Dans un premier temps donc, nous avons obtenu une sous-classe P_{d}^{k} de la classe $P_{\mathfrak{c}}^{k}$, une sous-classe qui est aussi vaste, puisque sa puissance est égale à $\mathfrak{c}$.

§ 14. Définition des fonctions déterminées par les arbres. Poids d'un arbre

Pour les fonctions déterminées il existe une méthode de définition plus suggestive que pour des fonctions quelconques de $P_{\mathfrak{c}}^{k}$, qui utilise l'appareil des arbres *).

Soient k et n des entiers, $N = k^n$. Considérons le schéma infini de la figure 4. Il est composé de *sommets* et d'*arêtes* orientées. Cette figure est un *arbre*. Le sommet ξ_0 s'appelle *racine de l'arbre*. Il en

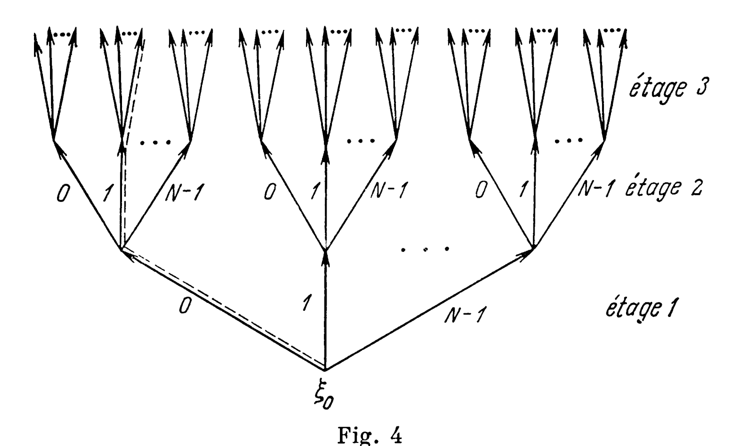


Fig. 4

part un faisceau de N arêtes formant le 1-*ier étage*. Chaque arête du premier étage mène à un sommet d'où partent N autres arêtes formant le 2-*ième étage*, etc. Les sommets qui sont les extrémités des arêtes du *m-ième étage* appartiennent aussi à ce dernier (le sommet ξ_0 est le sommet de l'étage zéro). Les arêtes de chaque faisceau sont numérotées de gauche à droite par les nombres $0, 1, \ldots, N-1$ (voir fig. 4) ou par leurs symboles dans un système de numération à base k:

$$\underbrace{(0, \ldots, 0, 0)}_{n}; \quad \underbrace{(0, \ldots, 0, 1)}_{n}; \quad \ldots; \quad \underbrace{(k-1, k-1, \ldots, k-1)}_{n}.$$

*) La définition rigoureuse d'un arbre est donnée dans la partie II.

Dans la suite on omettra d'écrire le numéro des arêtes sur les figures. On appellera *branche d'un arbre* un sous-ensemble connexe *) d'arêtes contenant exactement une arête dans chaque étage. Il est évident qu'à toute branche d'un arbre on peut associer une séquence

$$\alpha = \{\alpha(1), \alpha(2), \ldots, \alpha(m), \ldots\},$$

où i est le numéro de l'étage, $\alpha(i)$ le numéro de l'arête appartenant à cette branche. Ainsi par exemple à la branche représentée par la ligne en pointillé sur la figure 4 correspond la séquence $\{0, 1, N-1, \ldots\}$. Il est évident que $\alpha \in E_N^{\mathrm{c}}$. La réciproque est vraie aussi : à toute séquence α de E_N^{c} est associée une branche de l'arbre. On est donc en possession d'une correspondance biunivoque entre les branches de l'arbre et les éléments de l'ensemble E_N^{c}. On peut de ce fait utiliser l'arbre pour la représentation géométrique de l'ensemble E_N^{c}.

Soit $f(X)$ une fonction de P_{d}^N $(N = k^n)$ et $X = (x_1, \ldots, x_n)$. Utilisons la relation

$$f(X) \sim \{f_1(X_1), f_2(X_1, X_2), \ldots, f_m(X_1, X_2, \ldots, X_m), \ldots\}$$

pour associer à toute arête de l'arbre, à l'aide de $f(\widetilde{X})$, un nombre de E^k. Pour cela prenons une arête quelconque du m-ième étage $(m = 1, 2, \ldots)$ et considérons le chemin qui mène de la racine à cette arête (ce chemin peut également être défini comme un segment d'une branche quelconque passant par l'arête considérée). Il est évident que le chemin est défini de façon unique et peut être caractérisé par un cortège $\alpha(1), \alpha(2), \ldots, \alpha(m)$ de numéros des arêtes du chemin, comptées à partir de la racine. A l'arête initiale associons le m-ième terme de la séquence de sortie, c'est-à-dire le nombre $\gamma(m)$, où

$$\gamma(m) = f_m(\alpha(1), \ldots, \alpha(m)).$$

L'arbre obtenu sera appelé *arbre numéroté* (ou plus exactement *arbre à arêtes numérotées*).

Exemple 3. a) Pour la fonction $f_{\&}(x_1, x_2)$ on a $k = n = 2$, $N = 4$ et

$$\gamma(m) = f_m(X_1, X_2, \ldots, X_m) = f_m(X_m) = x_1(m) \& x_2(m).$$

Donc, $\gamma(m)$ dépend uniquement du dernier terme du cortège conduisant à l'arête donnée, c'est-à-dire du numéro de l'arête.

---

*) Un ensemble ordonné d'arêtes dans lequel l'extrémité d'une arête est l'origine de la suivante.

A l'arête de numéro $0 = (0, 0)$ correspond la valeur $0\&0=0$,
» » $1 = (0, 1)$ » » $0\&1=0$,
» » $2 = (1, 0)$ » » $1\&0=0$,
» » $3 = (1, 1)$ » » $1\&1=1$.

L'arbre numéroté correspondant est représenté sur la figure 5.

b) Pour la fonction $z = x + y$ on a $k = 2$, $n = 2$ et $N = 4$. Il est évident que

$$z(m) = \begin{cases} x(m)+y(m) \pmod{2} & \text{s'il n'y a pas de retenue,} \\ x(m)+y(m)+1 \pmod{2} & \text{s'il y a une retenue.} \end{cases}$$

Ceci nous suggère une règle pour obtenir un arbre numéroté (fig. 6). Le processus d'affectation de nombres aux arêtes commence par le

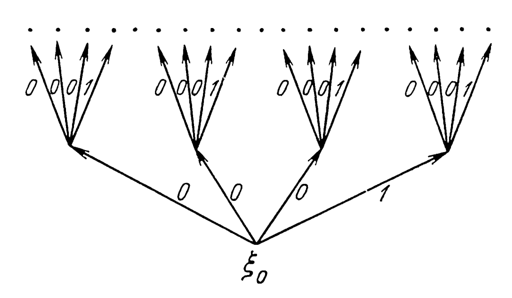


Fig. 5

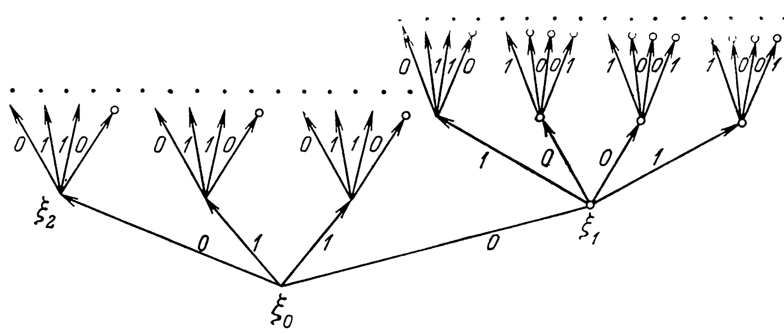


Fig. 6

premier étage. On passe au deuxième étage et ainsi de suite. Ceci étant, s'il y a une retenue, l'extrémité de l'arête correspondante est marquée d'un rond. Ceci permet d'effectuer les calculs dans l'étage suivant.

c) Pour la fonction $z = x^2$ on a $k = 2$, $n = 1$ et $N = 2$. L'expression de $z(m)$ par une fonction $f_m(x(1), \ldots, x(m))$ est bien plus

compliquée que dans les exemples précédents. Il est plus commode de calculer $z(m)$ à l'aide de l'algorithme d'élévation au carré. On voit que

$$z(1) = x(1),$$
$$z(2) = 0,$$
$$z(3) = x(2) + x(1)\,x(2) \pmod{2},$$
$$z(4) = x(1)\,x(3) + x(1)\,x(2) \pmod{2}, \quad \text{etc.}$$

Sur la figure 7 est représentée la portion initiale de l'arbre numéroté correspondant.

Nous voyons donc qu'une fonction déterminée nous donne un arbre numéroté. La réciproque est mise en défaut : un arbre numéroté peut définir plusieurs fonctions déterminées. Les paramètres N

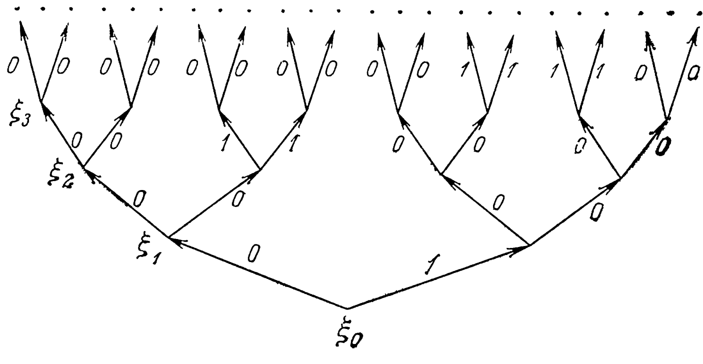


Fig. 7

et k' (où N est le nombre d'arêtes incidentes à chaque sommet vers l'extérieur, k' le maximum de nombres affectés aux arêtes) constituent une des solutions que peut admettre l'équation $N = k^n$ pour $k \geqslant k'$. (La solution $k = N$ et $n = 1$ existe toujours, c'est-à-dire qu'on peut toujours définir une fonction déterminée d'une variable.) Cependant, si l'on construit un arbre numéroté correspondant à une fonction déterminée $f(x_1, \ldots, x_n)$, cet arbre numéroté de paramètres k et n définit une seule fonction déterminée, en l'occurrence $f(x_1, \ldots, x_n)$. On peut donc utiliser les arbres numérotés pour étudier les fonctions déterminées.

Considérons un arbre numéroté pour une fonction déterminée $f(X) = f(x_1, \ldots, x_n)$. Soit ξ un sommet quelconque du m-ième étage de cet arbre. A partir de la racine ξ_0 on peut accéder à cet étage par le chemin

$$\alpha(1), \ldots, \alpha(m)$$

(pour $\xi = \xi_0$ ce chemin est vide). L'ensemble de toutes les branches issues de ξ engendre un arbre de racine ξ que nous appellerons *sous-arbre spécial* de l'arbre initial. Ce sous-arbre est défini par l'ensemble de toutes les séquences de E_k^c d'origine fixe

$$\alpha(1), \ldots, \alpha(m).$$

L'arbre initial étant numéroté, le sous-arbre l'est également. Si l'on numérote les étages du sous-arbre en commençant par le premier, il lui correspondra une fonction déterminée $f^{\xi}(X)$. Cette fonction peut être représentée analytiquement de la manière suivante: soit

$$f(X) \sim \{f_1(X_1), f_2(X_1, X_2), \ldots\},$$
$$f^{\xi}(X) \sim \{f_1^{\xi}(X_1), f_2^{\xi}(X_1, X_2), \ldots\}.$$

Alors

$$f_i^{\xi}(X_1, \ldots, X_i) = f_{m+i}(\alpha(1), \ldots, \alpha(m), X_1, \ldots, X_i)$$
$$(i = 1, 2, \ldots).$$

Définition. Deux sous-arbres de racines ξ_1 et ξ_2 d'un arbre sont dits *équivalents* si

$$f^{\xi_1}(X) \equiv f^{\xi_2}(X).$$

Il est évident que les numérotations de deux sous-arbres équivalents coïncident par une superposition. Ainsi, dans l'arbre de la figure 5 tous les sous-arbres sont équivalents, dans celui de la figure 6, les sous-arbres de sommets ξ_0 et ξ_2 le sont, mais pas ceux de sommets ξ_0 et ξ_1.

La relation d'équivalence permet de partager l'ensemble des sous-arbres d'un arbre en classes d'équivalence.

Définition. Le nombre r des classes d'équivalence de l'ensemble des sous-arbres d'un arbre s'appelle *poids de l'arbre* et respectivement *poids de la fonction déterminée* *).

Autrement dit, le poids est le nombre maximal de sous-arbres deux à deux non équivalents. Ceci n'exclut pas du reste le cas où le poids r est infini.

Revenons à l'exemple 3. Dans l'arbre de la fonction $f_{\&}(x_1, x_2)$ (fig. 5), on a vu que tous les sous-arbres étaient équivalents, donc $r = 1$. Dans l'arbre de la fonction $z = x + y$ (fig. 6) chaque sous-

---

*) Cette définition peut être étendue aux constantes. La séquence $\{\gamma(1), \gamma(2), \ldots, \gamma(m), \ldots\}$ est représentée par un arbre dégénéré constitué d'une seule branche

$$\circ \xrightarrow{\gamma(1)} \circ \xrightarrow{\gamma(2)} \circ \cdots \circ \xrightarrow{\gamma(m)} \circ \cdots$$

On peut considérer des sous-arbres et définir leur équivalence.

arbre est équivalent soit à un sous-arbre de racine ξ_0, soit à un sous-arbre de racine ξ_1, donc $r = 2$.

Dans l'arbre de la fonction $z = x^2$ (fig. 7) les sous-arbres de racines $\xi_0, \xi_1, \xi_2, \ldots$ (situés sur la branche de gauche) sont deux à deux non équivalents, puisqu'en vertu de la relation

$$\{\underbrace{0 \ldots 0}_{i} \alpha(i+1), \ldots\}^2 = \{\underbrace{0 \ldots 0}_{2i} \alpha(i+1), \ldots\}$$

le sous-arbre de racine ξ_i ne compte que des zéros dans les i premiers étages et le $(i+1)$-ième étage possède une arête à laquelle est affectée la valeur 1. Ici $r = \infty$.

Il est possible de numéroter les sommets d'arbres numérotés. Pour cela il faut d'abord numéroter les classes d'équivalence avec

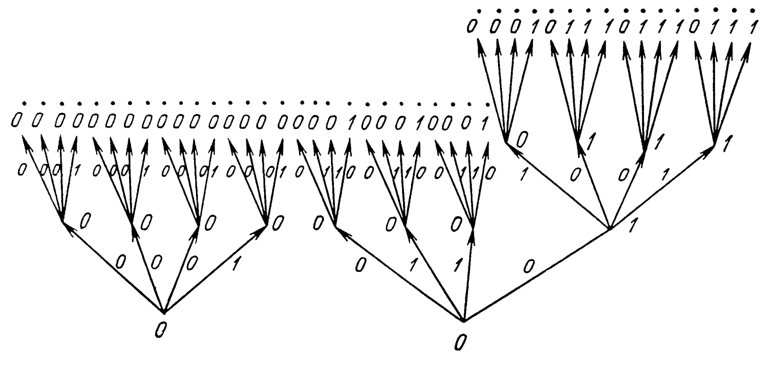


Fig. 8

les nombres 0, 1, . . . de telle sorte que la classe qui contient l'arbre de départ ait le numéro 0. Donc, le numérotage est entaché d'arbitraire. On prend ensuite un sommet quelconque ξ et on définit la classe contenant l'arbre de sommet ξ. Soit $\varkappa$ le numéro de cette dernière. Alors le sommet ξ a pour numéro $\varkappa$. On obtient un arbre dont les sommets sont aussi numérotés, le numéro de la racine étant 0. La figure 8 représente les numérotations des sommets pour les fonctions $f_{\&}(x_1, x_2)$ et $z = x + y$.

Si dans l'arbre considéré on ne conserve que la numérotation des sommets, il est immédiat de voir que la numérotation des arêtes n'est pas restituée de façon unique. Néanmoins, cette circonstance ne minimise pas la valeur de la numérotation des sommets pour l'étude de l'arbre. Dans bien des cas la numérotation des sommets peut être effectuée de pair avec celle des arêtes. Ainsi, dans l'exemple de la fonction $z = x + y$ le numéro du sommet 0 apparaît s'il n'y

a pas de retenue, le numéro 1 s'il y a retenue *). Considérons maintenant un arbre dont les arêtes et les sommets sont numérotés. Prenons une branche quelconque. Elle passe par les sommets

$$\xi_0,\ \xi_1,\ \ldots,\ \xi_i,\ \ldots,\ \xi_j,\ \ldots$$

Soient

$$0,\ \varkappa_1,\ \ldots,\ \varkappa_i,\ \ldots,\ \varkappa_j,\ \ldots$$

les numéros respectifs de ces sommets. Admettons que $\varkappa_i = \varkappa_j$ $(i \neq j)$ et que pour tous les couples $(i,\ j)$ $(i \neq j)$ tels que $\varkappa_i = \varkappa_j$,

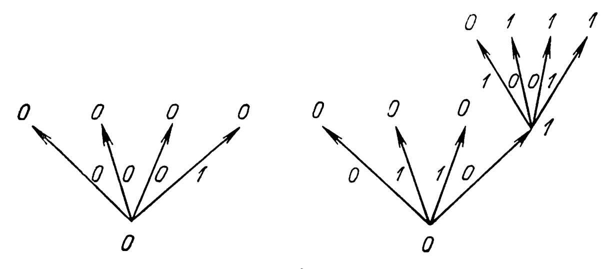


Fig. 9

l'indice j soit le plus petit. Tronquons cette branche et conservons sa portion initiale jusqu'au sommet ξ_j. En effectuant cette opération pour chaque branche on obtient un *arbre tronqué*.

Dans le cas d'une fonction de poids r fini, les numéros des sommets se répètent sur chaque branche et le numéro j qui définit la troncature vérifie l'inégalité $j \leqslant r$. Donc, pour de telles fonctions l'arbre tronqué sera fini.

La figure 9 représente des arbres tronqués pour les fonctions $f_{\&}(x_1,\ x_2)$ et $z = x + y$. Ces arbres tronqués s'obtiennent directement des arbres de la figure 8.

On voit immédiatement qu'un arbre tronqué à arêtes et sommets numérotés permet de restituer complètement l'arbre numéroté initial.

§ 15. Fonctions déterminées bornées et procédés de définition de ces fonctions

Définition. On appelle *fonction déterminée bornée* (en abrégé f.d.b.) une fonction déterminée de poids fini.

Désignons par P^k_{db} **) la classe des f.d.b.

*) On rappelle que dans l'exemple 3, b) on a marqué d'un rond l'extrémité de l'arête correspondante en cas de retenue.

**) La classe P^k_{db} contient notamment toutes les séquences périodiques de $E^{\text{c}}_{\varkappa}$.

Cette classe contient les fonctions des exemples 3, a) et 3, b) et, comme le montre l'exemple 3, c), est une sous-classe propre de la classe P_d^k des fonctions déterminées.

L'arbre numéroté (infini) de toute f.d.b. peut toujours être ramené à un arbre fini à arêtes et sommets numérotés. Si dans cet arbre tronqué on procède à une identification des sommets de mêmes numéros, on obtient ce qu'on appelle un *diagramme de Moore* *).

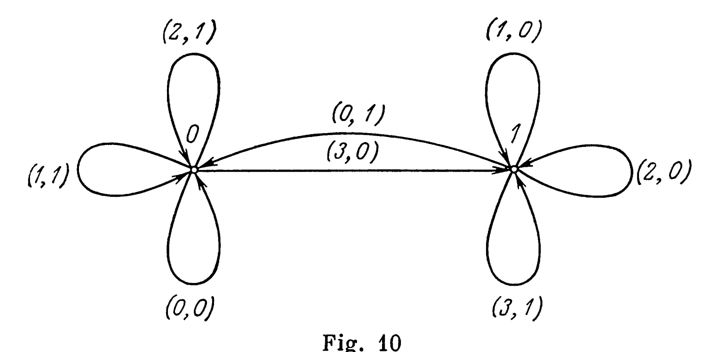


Fig. 10

La figure 10 représente un diagramme de Moore de la fonction $z = x + y$: le sommet initial est signifié par 0, et, pour la commodité, les arêtes sont affectées d'un couple (α, γ) de nombres dont le premier représente le numéro de l'arête et le second correspond à ladite arête.

Donc, une f.d.b. peut être définie aussi bien par des arbres numérotés infinis que par des diagrammes de Moore. Dans le cas général où f est de poids r, le diagramme de Moore possède r sommets dont l'un est signifié comme initial ; de chaque sommet partent $N = k^n$ arêtes ; aux arêtes sont affectés les couples $(0, \gamma')$, $(1, \gamma'')$, ..., $(N - 1, \gamma^{(N)})$. Dans la suite, tout diagramme doué de ces propriétés sera appelé diagramme de Moore. Les diagrammes de Moore permettent de construire des f.d.b. de tout poids. Dans ces constructions il faut avoir présent à l'esprit le fait que si une f.d.b. est restituée de façon unique par un diagramme de Moore donné formellement, celui-ci ne coïncide pas nécessairement avec le diagramme construit pour cette fonction par le procédé sus-mentionné. Ainsi, on s'aperçoit sans peine que le diagramme de la figure 11 définit la fonction $z = x + y$, mais qu'il diffère du diagramme de la figure 10.

---

*) Les diagrammes de Moore peuvent aussi servir à représenter des fonctions déterminées. Dans ce cas ils comportent en général une infinité de sommets.

Donc, si les diagrammes de Moore à r sommets ne représentent pas tous des f.d.b. de poids r, ils permettent du moins d'estimer le nombre de f.d.b. de poids r, dépendant des variables $x_1, \ldots, x_n$.

Théorème 2. *Le nombre $p(k, n, r)$ de f.d.b. de P_{c}^k de poids r, dépendant des n variables $x_1, \ldots, x_n$ est $\leqslant (rk)^{rk^n}$.*

Démonstration. Considérons un diagramme de Moore d'une f.d.b. de poids r. De chaque sommet il part $N = k^n$ arêtes, la α-ième

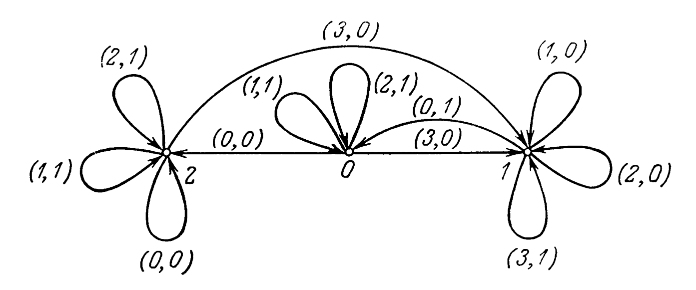


Fig. 11

arête étant reliée à l'un des r sommets et affectée du couple (α, γ), où $0 \leqslant \gamma \leqslant k - 1$. Donc, le nombre $p(k, n, r)$ est au plus égal à celui des diagrammes de Moore de la forme indiquée plus haut. Ces diagrammes peuvent être construits de la manière suivante.

Considérons r sommets numérotés par les nombres $0, \ldots, r - 1$ (0 est le sommet distingué). De chaque sommet partent $N = k^n$ arêtes numérotées par les nombres $0, \ldots, N - 1$. On a rN arêtes. Chacune d'elles peut être reliée à l'un quelconque des r sommets et être affectée de l'un quelconque des k nombres. Donc

$$p(k, n, r) \leqslant (rk)^{rN} = (rk)^{rk^n}.$$

C.q.f.d.

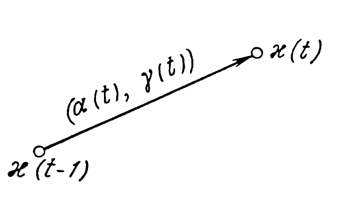


Fig. 12

Soit $f(X) = f(x_1, \ldots, x_n)$ une f.d.b. Considérons son diagramme de Moore. Supposons qu'à l'instant $t - 1$ on se soit trouvé au sommet $\varkappa(t - 1)$. Alors à l'arrivée du nombre $\alpha(t)$ à l'instant t on se déplace sur le diagramme le long de l'arête $\alpha(t)$ issue du sommet $\varkappa(t - 1)$ (cf. fig. 12). On obtient ainsi la valeur de sortie $\gamma(t)$ et l'on passe au sommet $\varkappa(t)$. Donc, les quantités $(\alpha(t), \varkappa(t - 1))$ définissent de façon unique la quantité $(\gamma(t), \varkappa(t))$.

Les quantités α et γ seront appelées respectivement *variables d'entrée* et *de sortie* et $\varkappa$, *état*.

Soient X, Q et Z des variables décrivant les valeurs respectivement d'une variable d'entrée α, d'un état $\varkappa$ et d'une variable de sortie γ.

Les raisonnements précédents nous conduisent aux équations suivantes *):

$$Z(t) = \mathcal{F}(X(t), Q(t-1)),$$
$$Q(t) = \mathcal{G}(X(t), Q(t-1)),$$

où $Q(0) = 0$. Ces équations sont appelées *équations canoniques*. Il est aisé de passer de l'écriture vectorielle de ces équations à l'écriture scalaire. Soit $l =]\log_k r[$. Alors

$$Z(t) = F'(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(t) = G'_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$q_l(t) = G'_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),$$
$$q_1(0) = \ldots = q_l(0) = 0.$$

F', G'_1, ..., G'_l sont des fonctions de P_k définies sur $\mathcal{E}$, sous-ensemble de $\underbrace{E^k \times \ldots \times E^k}_{n+l}$, plus exactement les variables x_1, ..., x_n prennent leurs valeurs dans E^k et le vecteur $(q_1, \ldots, q_l)$, r valeurs (par exemple les symboles binaires des nombres 0, 1, ..., $r-1$). Il est évident que l'ensemble $\mathcal{E}$ *est un cylindre* (ou un ensemble *cylindrique*) *par rapport à* $x_1, \ldots, x_n$ **).

En prolongeant les fonctions F', G'_1, ..., G'_l au domaine $\underbrace{E^k \times \ldots \times E^k}_{n+l}$ tout entier, on obtient respectivement les fonctions F, G_1, ..., G_l de P_k, et les équations canoniques prennent une forme plus commode (puisque l'on peut se servir de l'appareil des formules développé antérieurement):

$$\left.\begin{aligned} z(t) &= F(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),\\ q_1(t) &= G_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ q_l(t) &= G_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)),\\ q_1(0) &= \ldots = q_l(0) = 0. \end{aligned}\right\} \quad (*)$$

*) Ces constructions effectuées pour les constantes de P^k_{db} nous donnent des équations dans lesquelles ne figure pas la variable X.

**) On dit qu'un ensemble $\mathcal{E} \subset \underbrace{E^k \times \ldots \times E^k}_{n+l}$ est *cylindrique par rapport aux premières coordonnées* $x_1, \ldots, x_n$ si et seulement si deux combinaisons quelconques $(\zeta'_1, \ldots, \zeta'_n, \eta_1, \ldots, \eta_l)$ et $(\zeta''_1, \ldots, \zeta''_n, \eta_1, \ldots, \eta_l)$ appartiennent ou non simultanément à $\mathcal{E}$.

Il est évident que le domaine de définition $\mathcal{E}$ des fonctions F', G'_1, ..., G'_l est un ensemble cylindrique par rapport à $x_1, \ldots, x_n$.

Pour les f.d.b. décrites dans les exemples 3, a) et 3, b), les équations canoniques s'écrivent respectivement:

$$z(t) = x_1(t) \,\&\, x_2(t),$$

$$z(t) = x(t) + y(t) + q(t-1) \pmod{2},$$
$$q(t) = x(t)\,y(t) \vee x(t)\,q(t-1) \vee y(t)\,q(t-1),$$
$$q(0) = 0.$$

Exemple 4. Supposons qu'une f.d.b. $f(x)$ est définie par le diagramme de Moore de la figure 13. Les fonctions correspondantes $\mathcal{F}$

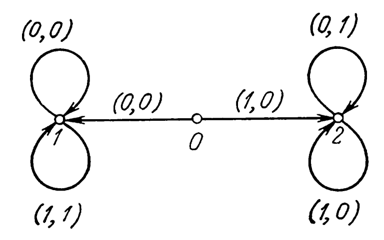


Fig. 13

et $\mathcal{G}$ sont représentées sur le tableau 14. En codant les états 0, 1, 2 par les combinaisons (0, 0), (0, 1) et (1, 0), on obtient les fonctions F', G_1', G_2' (voir tableau 15).

Tableau 14

X	Q	$\mathcal{F}$	$\mathcal{G}$
0	0	0	1
0	1	0	1
0	2	1	2
1	0	0	2
1	1	1	1
1	2	0	2

En prolongeant les fonctions F', G_1' et G_2', par exemple comme dans le tableau 16, on obtient les fonctions F, G_1 et G_2. Une fois en possession des fonctions F, G_1, G_2 on déduit les équations canoniques

$$z(t) = \bar{x}(t) \,\&\, q_1(t-1) \vee x(t) \,\&\, q_2(t-1),$$
$$q_1(t) = q_1(t-1) \vee x(t) \,\&\, \bar{q}_2(t-1),$$
$$q_2(t) = q_2(t-1) \vee \bar{x}(t) \,\&\, \bar{q}_1(t-1),$$
$$q_1(0) = q_2(0) = 0.$$

Tableau 15

x	q_1	q_2	F'	G'_1	G'_2
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	ne sont pas définies		
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	1	ne sont pas définies		

Tableau 16

x	q_1	q_2	F	G_1	G_2
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	1	1	1	1

On remarquera que pour $r = 1$ les variables q ne figurent pas dans les équations canoniques.

Donc, à toute f.d.b. on peut associer des équations canoniques. Mais ces équations ne sont pas uniques. Cela est dû:

a) aux divers procédés de codage (numérotation) des états;

b) aux divers procédés de prolongement des fonctions F', G'_1, ..., G'_l.

Il est immédiat de voir que les équations canoniques permettent de calculer la séquence de sortie

$$\gamma = \{\gamma(1), \gamma(2), \ldots\}$$

d'après la séquence d'entrée

$$\alpha = \{\alpha(1), \alpha(2), \ldots\}.$$

Pour définir une f.d.b. on est souvent conduit à étudier des systèmes arbitraires d'équations de type (*) (dans lesquels l peut être plus grand que $]\log_k r[$). En particulier, il est possible que les équations de ce type définissant une f.d.b. f ne coïncident avec aucune autre équation canonique associée à f.

§ 16. Opérations sur les f.d.b.

Pour définir les opérations sur les f.d.b. il est commode de le faire à partir des classes P^k_{c} et P^k_{d}.

On munit P^k_{c} de même que P_k de l'opération de superposition: on définit d'abord la notion de formule sur un système de fonctions de P^k_{c} et ensuite à toute formule on associe une fonction de P^k_{c}. Le théorème suivant est immédiat.

Théorème 3. *La classe des fonctions déterminées est fermée pour l'opération de superposition.*

Il est commode de représenter les fonctions déterminées sous forme de diagrammes fonctionnels. Si un système contient la fonction identique, la superposition se ramène à une application multiple de superpositions élémentaires de la forme

$$f(X) = f_0(f_1(X^1), \ldots, f_m(X^m)).$$

Il suffit donc d'indiquer la forme d'un diagramme fonctionnel pour une superposition élémentaire (voir fig. 14). Les carrés figurent les convertisseurs réalisant les fonctions mentionnées à l'intérieur.

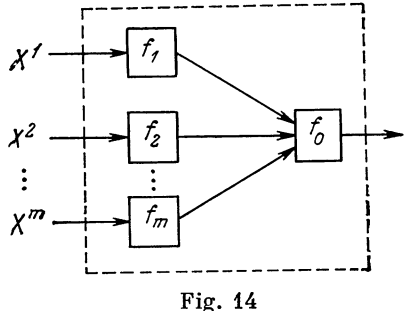


Fig. 14

Théorème 4. *La classe des f.d.b. est fermée pour la superposition.*

Démonstration. Comme la classe des f.d.b. contient la fonction identique, pour prouver ce théorème il suffit de montrer que la fonction $f(X)$ obtenue à partir des f.d.b. $f_0, f_1, \ldots, f_m$ à l'aide de la formule

$$f(X) = f_0(f_1(X^1), \ldots, f_m(X^m))$$

est une f.d.b.

Etant donné que, comme précédemment, la fonction est considérée aux variables inessentielles près (et que l'adjonction ou l'élimination de telles variables ne modifie pas le poids d'une f.d.b.), on peut admettre que les fonctions $f_1, \ldots, f_m$ dépendent des mêmes variables $x_1, \ldots, x_n$, c'est-à-dire que

$$f(x_1, \ldots, x_n) = f_0(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)).$$

Ecrivons les équations canoniques pour $f_0, f_1, \ldots, f_m$:

$$z_0(t) = F_0(y_1(t), \ldots, y_m(t), q_1^0(t-1), \ldots, q_{l_0}^0(t-1)),$$
$$q_1^0(t) = G_1^0(y_1(t), \ldots, y_m(t), q_1^0(t-1), \ldots, q_{l_0}^0(t-1)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$q_{l_0}^0(t) = G_{l_0}^0(y_1(t), \ldots, y_m(t), q_1^0(t-1), \ldots, q_{l_0}^0(t-1)),$$
$$q_1^0(0) = \ldots = q_{l_0}^0(0) = 0;$$
$$z_1(t) = F_1(x_1(t), \ldots, x_n(t), q_1^1(t-1), \ldots, q_{l_1}^1(t-1)),$$
$$q_1^1(t) = G_1^1(x_1(t), \ldots, x_n(t), q_1^1(t-1), \ldots, q_{l_1}^1(t-1)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$q_{l_1}^1(t) = G_{l_1}^1(x_1(t), \ldots, x_n(t), q_1^1(t-1), \ldots, q_{l_1}^1(t-1)),$$
$$q_1^1(0) = \ldots = q_{l_1}^1(0) = 0;$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$z_m(t) = F_m(x_1(t), \ldots, x_n(t), q_1^m(t-1), \ldots, q_{l_m}^m(t-1)),$$
$$q_1^m(t) = G_1^m(x_1(t), \ldots, x_n(t), q_1^m(t-1), \ldots, q_{l_m}^m(t-1)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$q_{l_m}^m(t) = G_{l_m}^m(x_1(t), \ldots, x_n(t), q_1^m(t-1), \ldots, q_{l_m}^m(t-1)),$$
$$q_1^m(0) = \ldots = q_{l_m}^m(0) = 0.$$

(On admet que les symboles q_j^i sont différents pour des couples (i, j) différents.) Montrons que f peut aussi être définie par des équations canoniques. En effet, posons

$$F(x_1, \ldots, x_n, q_1^0, \ldots, q_{l_0}^0, q_1^1, \ldots, q_{l_1}^1, \ldots, q_1^m, \ldots, q_{l_m}^m) =$$
$$= F_0(F_1(x_1, \ldots, x_n, q_1^1, \ldots, q_{l_0}^1), \ldots$$
$$\ldots, F_m(x_1, \ldots, x_n, q_1^m, \ldots, q_{l_m}^m), q_1^0, \ldots, q_{l_0}^0),$$

$$G_{ij}(x_1, \ldots, x_n, q_1^0, \ldots, q_{l_0}^0, q_1^1, \ldots, q_{l_1}^1, \ldots, q_1^m, \ldots, q_{l_m}^m) =$$
$$= \begin{cases} G_j^0(F_1(x_1, \ldots, x_n, q_1^1, \ldots, q_{l_1}^1), \ldots \\ \qquad \ldots, F_m(x_1, \ldots, x_n, q_1^m, \ldots, q_{l_m}^m), q_1^0, \ldots, q_{l_0}^0) \\ \qquad\qquad (i = 0,\ 1 \leqslant j \leqslant l_0), \\ G_j^i(x_1, \ldots, x_n, q_1^i, \ldots, q_{l_i}^i) \quad (i > 0,\ 1 \leqslant j \leqslant l_i). \end{cases}$$

Alors les équations

$$z(t) = F(x_1(t), \ldots, x_n(t), q_1^0(t-1), \ldots$$
$$\ldots, q_{l_0}^0(t-1), \ldots, q_1^m(t-1), \ldots, q_{l_m}^m(t-1)),$$
$$q_j^i(t) = G_{ij}(x_1(t), \ldots, x_n(t), q_1^0(t-1), \ldots$$
$$\ldots, q_{l_0}^0(t-1), \ldots, q_1^m(t-1), \ldots, q_{l_m}^m(t-1)),$$
$$q_j^i(0) = 0,$$
$$0 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant l_m,$$

définissent de toute évidence la fonction f. C.q.f.d.

Dans P_d^k définissons l'*opération R de retroaction.*

Définition. On dit qu'une fonction déterminée $f(x_1, \ldots, x_i, \ldots, x_n)$ *dépend avec retard de la variable* x_i si pour toutes séquences d'entrée

$$\alpha_1 = \{\alpha_1(1), \alpha_1(2), \ldots, \alpha_1(t), \ldots\},$$
$$\ldots\ldots\ldots\ldots$$
$$\alpha_n = \{\alpha_n(1), \alpha_n(2), \ldots, \alpha_n(t), \ldots\}$$

et pour tout instant t la valeur $\gamma(t)$, où

$$\gamma = f(\alpha_1, \ldots, \alpha_n),$$

est complètement définie par les valeurs des t premiers termes de la séquence

$$\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n$$

et par les valeurs des $t-1$ premiers termes de la séquence α_i (donc $\gamma(t)$ ne dépend pas de $\alpha_i(t)$).

Exemple 5. Considérons une fonction $f(x)$ de P_d^2 pour laquelle $\gamma(t) = \alpha(t-1)$ et $\gamma(1) = 0$, c'est-à-dire que $f(x)$ décale la séquence d'entrée d'un rang. Dans la suite on désignera cette fonction par $\vec{x}$. La figure 15 représente l'arbre correspondant à la fonction $\vec{x}$. On voit sur cette figure que $\vec{x}$ est une f.d.b. de poids 2 dont les équations canoniques sont:

$$z(t) = q(t-1),$$
$$q(t) = x(t),$$
$$q(0) = 0.$$

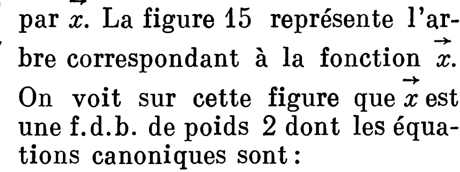


Fig. 15

On voit aussitôt que la fonction $\vec{x}$ dépend de x avec retard.

Supposons que $f(x_1, \ldots, x_n)$ dépend avec retard des variables $x_{i_1}, \ldots, x_{i_s}$. Il est alors évident que pour toutes séquences d'entrée

$$\alpha_1 = \{\alpha_1(1), \alpha_1(2), \ldots, \alpha_1(t), \ldots\},$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\alpha_n = \{\alpha_n(1), \alpha_n(2), \ldots, \alpha_n(t), \ldots\}$$

et pour tout instant t la valeur $\gamma(t)$, où

$$\gamma = f(\alpha_1, \ldots, \alpha_n),$$

est complètement définie par les valeurs des t premiers termes des séquences

$$\alpha_1, \ldots, \alpha_{i_1-1}, \alpha_{i_1+1}, \ldots, \alpha_{i_s-1}, \alpha_{i_s+1}, \ldots, \alpha_n$$

et par les valeurs des $t-1$ premiers termes des séquences

$$\alpha_{i_1}, \ldots, \alpha_{i_s}$$

(donc, $\gamma(t)$ ne dépend pas de $\alpha_{i_1}(t), \ldots, \alpha_{i_s}(t)$).

Lorsque $f(x_1, \ldots, x_n) \in P_{db}^k$, on peut définir la dépendance avec retard par rapport à la variable x_i $(1 \leqslant i \leqslant n)$ en termes

d'équations canoniques: $f(x_1, \ldots, x_n)$ dépend de x_i $(1 \leqslant i \leqslant n)$ avec retard si et seulement si elle peut être définie par les équations canoniques

$$\begin{aligned} z(t) &= F(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \\ q_1(t) &= G_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (*) \\ q_l(t) &= G_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \\ q_1(0) &= \ldots = q_l(0) = 0, \end{aligned}$$

dans lesquelles la fonction $F(x_1, \ldots, x_n, q_1, \ldots, q_l)$ en tant que fonction de P_k ne dépend pas essentiellement de x_i.

Dans l'exemple 5 on voit que $F(x, q) \equiv q$, c'est-à-dire que F ne dépend pas essentiellement de x, ainsi que le laissait prévoir la deuxième définition de la dépendance avec retard par rapport à une variable.

Supposons maintenant que la f.d.b. $f(x_1, \ldots, x_n)$ dépend avec retard des variables $x_{i_1}, \ldots, x_{i_s}$. Alors à partir de $(*)$ on peut trouver pour chaque x_{i_ν} $(\nu = 1, \ldots, s)$ des équations canoniques associées à f, de la forme

$$\begin{aligned} z(t) &= F_\nu(x_1(t), \ldots, x_{i_\nu - 1}(t), x_{i_\nu + 1}(t), \ldots \\ &\qquad \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \\ q_1(t) &= G_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ q_l(t) &= G_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \\ q_1(0) &= \ldots = q_l(0) = 0, \end{aligned}$$

où F_ν est le prolongement d'une fonction F' à P_k et ne dépend pas essentiellement de x_{i_ν}. Ainsi les prolongements

$$F_1, \ldots, F_s$$

sont en général différents de la fonction F'. Il se pose alors la question de savoir si on peut les choisir tels que

$$F_1 \equiv \ldots \equiv F_s \equiv F.$$

(Autrement dit, peut-on trouver un prolongement F de F' qui ne dépende pas essentiellement de $x_{i_1}, \ldots, x_{i_s}$.)

Ceci n'est pas valable pour toute fonction non partout définie de P_k, ainsi que le montre l'exemple suivant.

Exemple 6. Soit $F'(x_1, x_2)$ une fonction définie sur un ensemble $\mathscr{E}$, où $\mathscr{E} = \{(0, 0), (1, 1)\}$ et $\mathscr{E} \subset E^2 \times E^2$ (voir tableau 17).

Il est évident que $F_1(x_1, x_2) \equiv x_1$ et $F_2(x_1, x_2) \equiv x_2$ sont des

prolongements de F' et que F_1 et F_2 ne dépendent pas essentiellement de x_2 et de x_1 respectivement. Par ailleurs, il n'existe pas de prolongement de F' qui ne dépende pas essentiellement à la fois de x_1 et de x_2.

Tableau 17

x_1	x_2	$F'(x_1, x_2)$
0	0	0
1	1	1

Mais, sous certaines conditions qui sont réunies dans notre cas, on peut répondre par l'affirmative à la question posée.

Théorème 5. *Soit* $F'(x_1, \ldots, x_n, q_1, \ldots, q_l)$ *une fonction de* P_k *définie sur un ensemble* $\mathscr{E}$, *cylindrique par rapport à* $x_1, \ldots, x_n$. *Supposons que* F' *admet des prolongements* $F_1, \ldots, F_s$ *tels que* F_1 *ne dépende pas essentiellement de* x_{i_1}, F_2 *ne dépende pas essentiellement de* x_{i_2}, *etc., enfin* F_s *ne dépende pas essentiellement de* x_{i_s}. *Il existe alors un prolongement* F *qui ne dépend pas essentiellement des variables* $x_{i_1}, \ldots, x_{i_s}$.

Démonstration. Posons

$$F(x_1, \ldots, x_n, q_1, \ldots, q_l) = \begin{cases} F' \text{ sur } \mathscr{E} \\ 0 \text{ en dehors de } \mathscr{E}. \end{cases}$$

Montrons que F ne dépend pas essentiellement de $x_{i_1}, \ldots, x_{i_s}$. Soient $(\zeta_1', \ldots, \zeta_n', \eta_1, \ldots, \eta_l)$ et $(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l)$ deux combinaisons quelconques qui sont confondues pour toutes les variables sauf éventuellement pour $x_{i_1}, \ldots, x_{i_s}$. Comme l'ensemble $\mathscr{E}$ est cylindrique, ces combinaisons appartiennent ou n'appartiennent pas simultanément à $\mathscr{E}$. Dans le premier cas

$$F(\zeta_1', \ldots, \zeta_n', \eta_1, \ldots, \eta_l) = F'(\zeta_1', \ldots, \zeta_n', \eta_1, \ldots, \eta_l),$$
$$F(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l) = F'(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l).$$

Dans le second,

$$F(\zeta_1', \ldots, \zeta_n', \eta_1, \ldots, \eta_l) = F(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l) = 0.$$

Montrons que

$$F'(\zeta_1', \ldots, \zeta_n', \eta_j, \ldots, \eta_l) = F'(\zeta_1'', \ldots, \zeta_n'', \eta_1, \ldots, \eta_l).$$

En effet, dans le cas contraire il existerait deux combinaisons $(\zeta_1''', \ldots, \zeta_n''', \eta_1, \ldots, \eta_l)$ et $(\zeta_1^{IV}, \ldots, \zeta_n^{IV}, \eta_1, \ldots, \eta_l)$ voisines en l'une des coordonnées $x_{i_1}, \ldots, x_{i_s}$ (par exemple x_{i_ν}) pour lesquelles la condition $\zeta_i' = \zeta_i''$ entraîne que $\zeta_i''' = \zeta_i^{IV}$ et

$$F'(\zeta_1''', \ldots, \zeta_n''', \eta_1, \ldots, \eta_l) \neq F'(\zeta_1^{IV}, \ldots, \zeta_n^{IV}, \eta_1, \ldots, \eta_l).$$

Mais alors, F' ne pourrait être prolongée à une fonction de P_k qui ne dépendrait pas essentiellement de x_{i_ν}, ce qui est contraire à l'hypothèse.

Donc, F ne dépend pas essentiellement de $x_{i_1}, \ldots, x_{i_s}$ et le théorème est prouvé.

Passons maintenant à la définition de l'opération R.

Soit $\{f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)\}$ $(m \geqslant 2)$ un système de fonctions déterminées. Supposons que f_d dépend de la variable x_j avec retard. Si l'on traite ce système comme un convertisseur

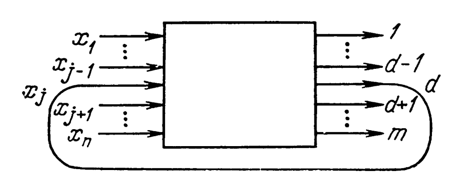


Fig. 16

à n entrées et m sorties, on peut alors relier la sortie d à l'entrée j (voir fig. 16), c'est-à-dire introduire une « rétroaction » entre la sortie d et l'entrée j. On obtient ainsi un convertisseur qui réalise le système de $m - 1$ fonctions déterminées

$\{f'_1(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n), \ldots, f'_{d-1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n),$
$f'_{d+1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n), \ldots, f'_m(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)\}$

dépendant des $n - 1$ variables $x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n$. Les fonctions $f'_1, \ldots, f'_{d-1}, f'_{d+1}, \ldots, f'_n$ se définissent formellement comme suit: soit α' une séquence d'entrée pour $x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n$.

1) Considérons $\alpha(1) = \{\alpha'_1(1), \ldots, \alpha'_{j-1}(1), 0, \alpha'_{j+1}(1), \ldots, \alpha'_n(1)\}$ et calculons sur cette combinaison la valeur $\gamma_d(1)$ de la fonction f_d à l'instant 1.

Considérons $\tilde{\alpha}(1) = \{\alpha'_1(1), \ldots, \alpha'_{j-1}(1), \gamma_d(1), \alpha'_{j+1}(1), \ldots, \alpha'_n(1)\}$ et calculons les valeurs des fonctions $f'_1, \ldots, f'_{d-1}, f'_{d+1}, \ldots, f'_m$ sur cette combinaison à l'instant 1.

2) Considérons $\alpha(2) = \{\alpha'_1(2), \ldots, \alpha'_{j-1}(2), \gamma_d(1), \alpha'_{j+1}(2), \ldots, \alpha'_n(2)\}$. Calculons la valeur $\gamma_d(2)$ de f_d sur les combinaisons $\alpha(1)$ et $\alpha(2)$ à l'instant 2.

Considérons $\tilde{\alpha}(2) = \{\alpha'_1(2), \ldots, \alpha'_{j-1}(2), \gamma_d(2), \alpha'_{j+1}(2), \ldots, \alpha'_n(2)\}$ et calculons les valeurs des fonctions $f'_1, \ldots, f'_{d-1}, f'_{d+1}, \ldots, f'_m$ sur les combinaisons $\tilde{\alpha}(1)$ et $\tilde{\alpha}(2)$ à l'instant 2, et ainsi de suite.

t) Considérons $\alpha(t) = \{\alpha'_1(t), \ldots, \alpha'_{j-1}(t), \gamma_d(t-1), \alpha'_{j+1}(t), \ldots, \alpha'_n(t)\}$ et calculons la valeur $\gamma_d(t)$ de la fonction f_d sur les combinaisons $\alpha(1), \alpha(2), \ldots, \alpha(t)$ à l'instant t.

Considérons $\tilde{\alpha}(t) = \{\alpha'_1(t), \ldots, \alpha'_{j-1}(t), \gamma_d(t), \alpha'_{j+1}(t), \ldots, \alpha'_n(t)\}$ et calculons les valeurs des fonctions $f'_1, \ldots, f'_{d-1}, f'_{d+1}, \ldots, f'_m$

sur les combinaisons $\tilde{\alpha}(1)$, $\tilde{\alpha}(2)$, ..., $\tilde{\alpha}(t)$ à l'instant t, et ainsi de suite.

Si $f_1, \ldots, f_m$ sont des f.d.b., l'opération R peut être définie par des équations canoniques. Supposons que f_d dépend avec retard de la variable x_j. Considérons le système d'équations canoniques pour $f_1, \ldots, f_m$:

$$
\begin{aligned}
z_1(t) &= F_1(x_1(t), \ldots, x_j(t), \ldots, x_n(t), q_1(t-1), \ldots \\
&\qquad \ldots, q_l(t-1)), \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
z_d(t) &= F_d(x_1(t), \ldots, -, \ldots, x_n(t), q_1(t-1), \ldots \\
&\qquad \ldots, q_l(t-1)), \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
z_m(t) &= F_m(x_1(t), \ldots, x_j(t), \ldots, x_n(t), q_1(t-1), \ldots \\
&\qquad \ldots, q_l(t-1)), \\
q_1(t) &= G_1(x_1(t), \ldots, x_j(t), \ldots, x_n(t), q_1(t-1), \ldots, \\
&\qquad q_l(t-1)), \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
q_l(t) &= G_l(x_1(t), \ldots, x_j(t), \ldots, x_n(t), q_1(t-1), \ldots \\
&\qquad \ldots, q_l(t-1)), \\
q_1(0) &= \ldots = q_l(0) = 0.
\end{aligned}
$$

Le trait dans la deuxième formule signifie que la fonction F_d ne dépend pas essentiellement de x_j. En éliminant de ce système d'équations canoniques la ligne d et en remplaçant x_j par F_d on obtient le nouveau système d'équations:

$$
\begin{aligned}
z_1(t) &= F_1(x_1(t), \ldots, F_d(x_1(t), \ldots, -, \ldots, x_n(t), \\
&\qquad q_1(t-1), \ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots \\
&\qquad \ldots, q_l(t-1)), \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
z_{d-1}(t) &= F_{d-1}(x_1(t), \ldots, F_d(x_1(t), \ldots, -, \ldots, x_n(t), \\
&\qquad q_1(t-1), \ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots \\
&\qquad \ldots, q_l(t-1)), \\
z_{d+1}(t) &= F_{d+1}(x_1(t), \ldots, F_d(x_1(t), \ldots, -, \ldots, x_n(t), \\
&\qquad q_1(t-1), \ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots \\
&\qquad \ldots, q_l(t-1)), \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots
\end{aligned}
$$

$$z_m(t) = F_m(x_1(t), \ldots, F_d(x_1(t), \ldots, -, \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots \ldots, q_l(t-1)),$$

$$q_1(t) = G_1(x_1(t), \ldots, F_d(x_1(t), \ldots, -, \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots \ldots, q_l(t-1)),$$

. . . .

$$q_l(t) = G_l(x_1(t), \ldots, F_d(x_1(t), \ldots, -, \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \ldots, x_n(t), q_1(t-1), \ldots \ldots, q_l(t-1)),$$

$$q_1(0) = \ldots = q_l(0) = 0.$$

Ce système d'équations définit visiblement les mêmes f.d.b. $f'_1, \ldots \ldots, f'_{d-1}, f'_{d+1}, \ldots, f'_m$ que celles définies plus haut. Il s'ensuit de même que le système de f.d.b. obtenu ne dépend pas du choix des équations canoniques pour $f_1, \ldots, f_m$ (sous réserve que F_d ne dépende pas essentiellement de x_j).

Exhibons un exemple montrant comment agit l'opération R dans ce cas. Pour système initial de f.d.b. prenons le système défini par les équations canoniques

$$z(i) = x(i) + y(i) + q(i-1) \pmod{2},$$
$$w(i) = x(i)\,y(i) \vee x(i)\,q(i-1) \vee y(i)\,q(i-1),$$
$$q(i) = y'(i), \quad q(0) = 0.$$

Comme indiqué plus haut, les fonctions z et w dépendent de la variable y' avec retard. Introduisons une rétroaction à l'aide de l'identité

$$w(i) = y'(i).$$

Par élimination on obtient les équations canoniques suivantes:

$$z(i) = x(i) + y(i) + q(i-1),$$
$$q(i) = x(i)\,y(i) \vee x(i)\,q(i-1) \vee y(i)\,q(i-1), \; q(0) = 0.$$

Donc, le résultat de l'opération R est une f.d.b. qui est la somme de deux séquences.

Théorème 6. *La classe des f.d.b. est fermée pour l'opération R.*

Soit $f_1, \ldots, f_m$ un système de fonctions déterminées dans lequel f_{d_1} et f_{d_2} dépendent avec retard respectivement de x_{j_1} et x_{j_2} ($j_1 \neq j_2$, $d_1 \neq d_2$, $m \geqslant 3$). On peut d'abord appliquer la rétroaction (d_1, j_1) et ensuite la rétroaction (d_2, j_2). On obtient en définitive un système

de $m-2$ fonctions déterminées dépendant de $n-2$ variables. Si l'on introduit les rétroactions dans un ordre différent : d'abord (d_2, j_2) et ensuite (d_1, j_1), on obtient un système qui, en principe, est différent du premier. Si $f_1, \ldots, f_m$ est un système de f.d.b., ceci apparaît plus facilement lorsqu'on construit des systèmes d'équations pour chacun des deux procédés d'introduction des rétroactions. Dans le premier cas on a substitué à la variable x_{j_1} une expression de la forme

$$F_{d_1}(\ldots, \overset{j_1}{-}, \ldots, \overset{j_2}{F_{d_2}}(\ldots F_{d_1}\ldots), \ldots),$$

et à x_{j_2}, l'expression

$$F_{d_2}(\ldots, \overset{j_1}{F_{d_1}}, \ldots, \overset{j_2}{-}, \ldots).$$

Dans le second cas, on a substitué à la variable x_{j_1} une expression de la forme

$$F_{d_1}(\ldots, \overset{j_1}{-}, \ldots, \overset{j_2}{F_{d_2}}, \ldots),$$

et à x_{j_2}, l'expression

$$F_{d_2}(\ldots, \overset{j_1}{F_{d_1}}(\ldots, F_{d_2}\ldots)), \ldots, \overset{j_2}{-}, \ldots).$$

On obtient donc généralement des systèmes différents qui définissent des systèmes différents de f.d.b.

Théorème 7. *Si un système de f.d.b. $f_1, \ldots, f_m$ contient des fonctions $f_{d_1}, \ldots, f_{d_s}$ ($m \geqslant s+1$; les indices sont deux à deux distincts) dépendant chacune avec retard des variables $x_{j_1}, \ldots, x_{j_s}$ (les indices sont deux à deux distincts), alors le système de fonctions obtenu par introduction des rétroactions $(d_1, j_1), \ldots, (d_s, j_s)$ ne dépend pas de l'ordre d'introduction de ces rétroactions.*

Démonstration. Il suffit de toute évidence de considérer le cas $s=2$. En se servant du théorème 5 on voit que F_{d_1} et F_{d_2} ne dépendent pas essentiellement de x_{j_1} et x_{j_2}. Donc, pour chaque ordre d'introduction des rétroactions, dans le système d'équations la variable x_{j_1} sera remplacée par F_{d_1} et la variable x_{j_2} par F_{d_2}, c'est-à-dire que les deux systèmes sont confondus (voir raisonnements précédant le théorème).

Dans la suite, l'application répétée de l'opération R nous placera en principe dans la situation décrite dans le théorème.

Dans les hypothèses du théorème, l'introduction répétée des rétroactions $(d_1, j_1), \ldots, (d_s, j_s)$ peut être représentée comme sur la figure 17, car sur cette figure les rétroactions ne sont pas ordonnées. D'autre part, l'introduction répétée des rétroactions $(d_1, j_1), \ldots$

. . ., (d_s, j_s) pour le système de f.d.b. $f_1, \ldots, f_m$ nous conduit à un système de $m - s$ fonctions de $n - s$ variables qui peuvent être

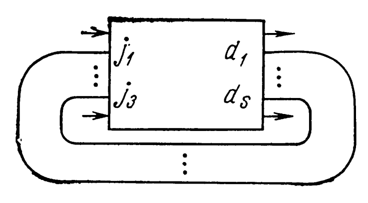


Fig. 17

calculées par une procédure non itérée du type de celle de la page 87, c'est-à-dire par un calcul simultané sur les variables $x_{j_1}, \ldots$ $\ldots, x_{j_s}$ *).

Les théorèmes 4 et 6 impliquent le

Théorème 8. *La classe P^k_{db} est fermée pour les opérations S et R.*

§ 17. Exemples de systèmes complets

On se propose d'étudier la complétude du système fonctionnel (P^k_{db}, S, R) construit dans le paragraphe précédent. Pour cela il est nécessaire d'étudier le lien entre les systèmes fonctionnels (P^k_{db}, S, R) et (P_k, S) et de mettre au point des procédures de construction de « formules » exprimant des f.d.b. au moyen de f.d.b. plus simples.

Nous avons vu au § 13 qu'à toute fonction Φ de P_k était associée une f.d.b. f_Φ. La correspondance

$$\Phi \to f_\Phi$$

engendre une application de P_k sur un sous-ensemble P^k de f.d.b.

*) Cette remarque est valable pour le cas aussi où $f_1, \ldots, f_m$ est un système de fonctions déterminées dans lequel les fonctions $f_{d_1}, \ldots, f_{d_s}$ dépendent avec retard des variables $x_{j_1}, \ldots, x_{j_s}$.

Introduisons une écriture analytique pour l'opération de rétroaction par analogie avec la représentation de la rétroaction pour les convertisseurs. Si $f_1, \ldots, f_m$ est un système de f.d.b. et $f_{d_1}, \ldots, f_{d_s}$ dépendent avec retard des variables $x_{j_1}, \ldots, x_{j_s}$, alors on représentera les rétroactions $(d_1, j_1), \ldots, (d_s, j_s)$ par

$$\begin{aligned} z_1 &= f_1(x_1, \ldots, x_n), \\ &\cdots\cdots\cdots \\ x_{j_1} &= f_{d_1}(x_1, \ldots, x_n), \\ &\cdots\cdots\cdots \\ x_{j_s} &= f_{d_s}(x_1, \ldots, x_n), \\ &\cdots\cdots\cdots \\ z_m &= f_m(x_1, \ldots, x_n). \end{aligned}$$

Supposons que Φ est une superposition Φ de P_k engendrée par une formule $\mathfrak{A}$, plus exactement

$$\Phi = \mathfrak{A}(\Phi_1, \ldots, \Phi_m).$$

Considérons les f.d.b. f_Φ, f_{Φ_1}, . . ., f_{Φ_m}, images par cette application des fonctions Φ, Φ_1, . . ., Φ_m. Il est immédiat de voir que

$$\mathfrak{A}(f_{\Phi_1}, \ldots, f_{\Phi_m}) = f_{\mathfrak{A}(\Phi_1, \ldots, \Phi_m)} = f_\Phi,$$

c'est-à-dire que l'image de la superposition engendrée par la formule $\mathfrak{A}$ est la superposition des images engendrées par la même formule $\mathfrak{A}$. Dans le langage des convertisseurs cette proposition admet

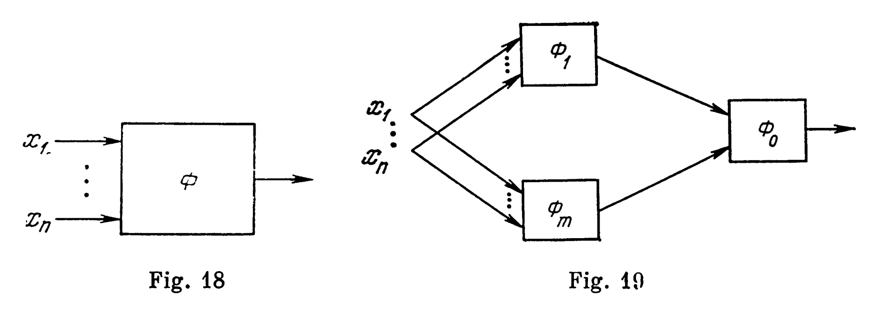


Fig. 18 Fig. 19

une interprétation assez simple. Supposons qu'un convertisseur (cf. fig. 18) réalise une fonction $\Phi(x_1, \ldots, x_m)$ de P_k. Si aux entrées de ce convertisseur on applique des valeurs quelconques aux instants $t = 1, 2, \ldots$, on aura de toute évidence à la sortie une réalisation de la f.d.b. $f_\Phi(x_1, \ldots, x_n)$. Supposons de façon analogue que le diagramme fonctionnel (voir fig. 19) réalise la superposition

$$\Phi(x_1, \ldots, x_n) = \Phi_0(\Phi_1(x_1, \ldots, x_n), \ldots, \Phi_m(x_1, \ldots, x_n)),$$

où les fonctions Φ_0, Φ_1, . . ., Φ_m appartiennent à P_k. Si aux entrées de ce diagramme fonctionnel on applique des valeurs aux instants $i = 1, 2, \ldots$, on obtiendra en sortie une réalisation de la f.d.b.

$$f_{\Phi_0(\Phi_1, \ldots, \Phi_m)} = f_\Phi.$$

Donc, l'application de P_k sur P^k est biunivoque et préserve la superposition.

Les systèmes fonctionnels (P_k, S) et (P^k, S) doués des propriétés mentionnées s'appellent *isomorphes*. L'isomorphisme permet d'étendre tous les résultats concernant (P_k, S) à (P^k, S). En particulier, il s'ensuit que P^k possède une base finie. Pour bases dans P^k on peut prendre par exemple

1) $\{f^0, f^1, \ldots, f^{k-1}, f_{I_0(x)}, \ldots, f_{I_{k-1}(x)}, f_{\min(x_1, x_2)}, f_{\max(x_1, x_2)}\}$, où $f^0, f^1, \ldots, f^{k-1}$ sont les images des constantes $0, 1, \ldots, k-1$;

2) $\{f_{V(x_1, x_2)}\}$, où $V(x_1, x_2)$ est la fonction de Webb.

Il est important de connaître la représentation de f.d.b. quelconques en termes d'opérations R et S par des f.d.b. plus simples (analogue du théorème de développement des fonctions de P_2 et de P_k, $k \geqslant 3$). On sait que toute f.d.b. peut être définie par des équations canoniques :

$$\begin{aligned} z(t) &= F(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \\ q_1(t) &= G_1(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ q_l(t) &= G_l(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_l(t-1)), \\ q_1(0) &= \ldots = q_l(0) = 0. \end{aligned}$$

Ces équations « expriment » la fonction f à l'aide de fonctions de P_k et ne sont pas des « formules » dans le système (P^k_{db}, S, R). Cependant, on passe sans peine de ces équations à une formule du système (P^k_{db}, S, R). Considérons l'expression

$$\begin{aligned} z &= f_F(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l), \\ q_1 &= f_{G_1}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l), \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ q_l &= f_{G_l}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l). \end{aligned}$$

Il est aisé de voir qu'elle est constituée des images de fonctions f_Φ (appartenant à P^k) et de $\vec{x}$ par les opérations S et R : d'abord dans les f.d.b. $f_F, f_{G_1}, \ldots, f_{G_l}$ les l dernières variables ont été remplacées par les f.d.b. $\vec{q}_1, \ldots, \vec{q}_l$ (opération de superposition). Les fonctions

$$f_F(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l),\ f_{G_1}(x_1, \ldots, x_n, \vec{q}_1, \ldots \\ \ldots, \vec{q}_l),\ \ldots,\ f_{G_l}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l)$$

dépendent avec retard des variables $q_1, \ldots, q_l$, donc il est possible d'introduire des rétroactions (qui, en vertu du théorème 7, sont indépendantes de l'ordre) et le résultat de l'application de l'opération R peut être écrit sous la forme de l'expression donnée. Donc, cette expression est une « formule » dans le système (P^k_{db}, S, R).

Ainsi, toute f.d.b. f peut être représentée par une « formule » (que nous appellerons aussi *formule canonique*) faisant intervenir des f.d.b. plus simples et les opérations S et R.

Théorème 9. *Le système de f.d.b.*

$$\{f^0, \ldots, f^{k-1}, f_{I_0}(x), \ldots, f_{I_{k-1}}(x), f_{\min}(x_1, x_2), f_{\max}(x_1, x_2), \vec{x}\}$$

est complet dans (P^k_{db}, S, R).

Démonstration. Soit $f(x_1, \ldots, x_n)$ une fonction arbitraire de P^k_{db}. Représentons-la par la formule canonique

$$z = f_F(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l),$$
$$q_1 = f_{G_1}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$
$$q_l = f_{G_l}(x_1, \ldots, x_n, \vec{q}_1, \ldots, \vec{q}_l).$$

Les fonctions $f_V, f_{G_1}, \ldots, f_{G_l}$ appartiennent à l'ensemble P^k et sont engendrées (voir corollaire de la page 92) par le système

$$\{f^0, \ldots, f^{k-1}, f_{I_0}(x), \ldots, f_{I_{k-1}}(x), f_{\min}(x_1, x_2), f_{\max}(x_1, x_2)\}.$$

Donc, la formule canonique peut être représentée par des formules du système initial. C.q.f.d.

On prouve de façon analogue le

Théorème 10. *Le système de f.d.b.* $\{f_V(x_1, x_2), \vec{x}\}$ *est complet dans* (P^k_{db}, S, R).

Théorème 11. *Soit* $\{\Phi_1, \ldots, \Phi_m\}$ *un système complet dans* (P_k, S). *Alors le système de f.d.b.*

$$\{f_{\Phi_1}(x_1, \ldots, x_n), \ldots, f_{\Phi_m}(x_1, \ldots, x_n), \vec{x}\}$$

est complet dans (P^k_{db}, S, R).

La proposition suivante généralise un résultat de V. Koudriavtsev. [3].

Théorème 12. *Il existe une f.d.b. f (analogue de la fonction de Sheffer) telle que le système* $\{f\}$ *constitué de la seule fonction f est complet dans* (P^k_{db}, S, R).

Démonstration. Soit $F_0(x_1, x_2, x_3, x_4)$ une fonction de P_k définie par la formule

$$F_0(x_1, x_2, x_3, x_4) = \max\,[x_1 \cdot x_4 + x_3(1 - x_4), x_2] + 1 \pmod{k}$$

(les opérations $\cdot$, $+$ et $-$ sont effectuées mod. k). Considérons la f.d.b.

$$f(x_1, x_2, x_1, x_4) = f_{F_0}(x_1, x_2, x_1, \vec{x}_4).$$

Montrons que le système $\{f\}$ est complet dans (P^k_{db}, S, R). Posons $x_1 = x_3$ et considérons la fonction

$$F_0(x_1, x_2, x_1, x_4) = \max\,[x_1, x_2] + 1 \pmod{k} = V(x_1, x_2),$$

où V est la fonction de Webb. En vertu de l'isomorphisme mentionné plus haut

$$f(x_1, x_2, x_1, x_4) = f_V(x_1, x_2).$$

La fonction $f_V(x_1, x_2)$, en engendrant P^k tout entier, permet de construire les fonctions $f^0, f^1, \ldots, f^{k-1}$ et $f_{x+k-1}(x)$. Considérons la superposition

$$f_{x+k-1}(f(f^1, f^0, f^0, x_4)) = f_{x+k-1}(f_{F_0}(f^1, f^0, f^0, \vec{x}_4)) =$$
$$= f_{F_0+k\cdot 1}(f^1, f^0, f^0, \vec{x}_4) = f_{F_0(1,\,0,\,0,\,x_4)+k-1}(\vec{x}_4) = f_{x_4}(\vec{x}_4) = \vec{x}_4.$$

Donc, on a obtenu $f_V(x_1, x_2)$ et $\vec{x}$ à partir de $\{f\}$ par des superpositions, ce qui en vertu du théorème 10 signifie que le système $\{f\}$ est complet. C.q.f.d.

Nous constatons que certaines propriétés du système (P_k, S) restent en vigueur pour le système (P^k_{db}, S, R). Pour les autres propriétés du système (P^k_{db}, S, R) il est difficile de se prononcer, car d'une part l'être fonctionnel P^k_{db} est bien plus compliqué que P_k et d'autre part P^k_{db} est muni de l'opération R. La première circonstance a tendance à détruire les propriétés positives, la deuxième, au contraire, à les renforcer. Il est difficile de dire à priori laquelle de ces tendances prend le pas sur l'autre. En effet, dans le problème de complétude, la complexité de l'être fonctionnel « prédomine » tout de même l'opération supplémentaire R. Pour illustrer ceci citons deux résultats sans les prouver. L'énoncé du premier d'entre eux peut être compris de manière intuitive, car on y utilise la notion d'algorithme.

Théorème 13 (M. Kratko [6, 7]). *Il n'existe pas d'algorithme permettant de dire si un système fini quelconque de f.d.b. est complet ou non.*

Théorème 14 (V. Koudriavtsev [4]). *L'ensemble des classes précomplètes dans (P^k_{db}, S, R) a la puissance du continu.*

Donc, le problème de la complétude de (P^k_{db}, S, R) soulève de grosses difficultés.

§ 18. Sur le lien entre les opérations S et R

Le système fonctionnel (P^k_{db}, S, R) possède de nombreuses propriétés spécifiques, dont l'énoncé et l'exposé nécessitent une bien plus grande place que pour (P_k, S) par exemple. Aussi, aborderons-nous ici un seul problème lié au fait que contrairement aux systèmes considérés, le système (P^k_{db}, S, R) contient deux opérations:

S et R. Il s'agira de savoir jusqu'à quel point ces opérations sont essentielles. Plus exactement: peut-on exprimer le résultat d'une opération par l'autre et en particulier y a-t-il une différence entre les opérations de fermeture dans les systèmes (P^k_{db}, S), (P^k_{db}, R) et (P^k_{db}, S, R)?

Montrons tout d'abord quelques propositions auxiliaires.

Théorème 15. *Soient $X = (x_1, \ldots, x_n)$, $f(X)$ une f.d.b. de poids r, α une séquence périodique de période p. Alors il existe r_1 $(r_1 \leqslant r)$ tel que la séquence γ $(\gamma = f(\alpha))$ soit périodique de période p_1, où $p_1 = r_1 p$.*

Démonstration. La séquence α étant périodique, il existe un t_0 tel que pour $t \geqslant t_0 \geqslant 1$

$$\alpha(t' + p) = \alpha(t').$$

La fonction $f(X)$ peut être définie par un diagramme de Moore de r sommets. Considérons l'axe numérique t et à chaque point t^* $(t^* =$

Fig. 20

Fig. 21

$= 1, 2, \ldots)$ associons deux nombres: le nombre $\alpha(t^*)$ et le nombre $\varkappa(t^* - 1)$ qui représente le numéro du sommet auquel on accède en partant de l'origine et en suivant le chemin $(\alpha(1), \ldots, \alpha(t^* - 1))$ (voir fig. 20). Considérons ensuite un réseau d'origine t_0 et de période p (cf. fig. 21).

Le diagramme de Moore contenant r sommets, deux au moins des nombres de la séquence

$$\varkappa(t_0 - 1),\ \varkappa(t_0 + p - 1),\ \ldots,\ \varkappa(t_0 + rp - 1)$$

sont confondus. Supposons que i et j sont tels que

$$\varkappa(t_0 + ip - 1) = \varkappa(t_0 + jp - 1) \quad (j > i \geqslant 0).$$

Posons $r_1 = j - i$. Il est évident que $r_1 \leqslant r$. Considérons un surréseau de ce réseau, de période $p_1 = r_1 p$ et d'origine $t_0 + ip$ (cf

fig. 22). Il est clair que les sommets $t_0 + ip$ et $t_0 + ip + p_1$ sont caractérisés par le fait que les valeurs α et $\varkappa$ y sont confondues. Cela signifie qu'en ces instants on se trouve dans un même sommet du

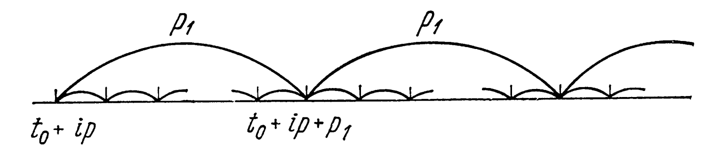


Fig. 22

diagramme et l'on se déplace ensuite sur la même arête. Donc, à l'instant suivant, on accède au même sommet, c'est-à-dire que les valeurs

$$\varkappa\,(t_0 + ip), \quad \varkappa\,(t_0 + jp)$$

sont également confondues. De plus, α étant périodique, on a

$$\alpha\,(t_0 + ip + 1) = \alpha\,(t_0 + jp + 1),$$

etc. D'où il résulte que les séquences de sortie

$$\{\gamma\,(t_0 + ip),\ \gamma\,(t_0 + ip + 1),\ \ldots,\},$$
$$\{\gamma\,(t_0 + jp),\ \gamma\,(t_0 + jp + 1),\ \ldots\}$$

sont confondues, c'est-à-dire que pour $t \geqslant t_0 + ip$

$$\gamma\,(t + p_1) = \gamma\,(t).$$

Ce que nous voulions.

Théorème 16. *Soient $f\,(X) = f_0\,(f_1\,(X), \ldots, f_m\,(X))$, où f_0, $f_1, \ldots, f_m$ sont des f.d.b. de poids respectifs $r_0, r_1, \ldots, r_m$ ne contenant que des facteurs premiers non supérieurs à R, α une séquence périodique dont la période ne comprend que des facteurs premiers, non supérieurs chacun à R. Alors $\gamma = f\,(\alpha)$ est une séquence périodique dont la période ne contient que des facteurs premiers, non supérieurs chacun à R.*

Démonstration. Soit α une sequence périodique dont la période p ne contient que des facteurs premiers non supérieurs à R. Considérons les séquences

$$\gamma_1 = f_1\,(\alpha), \ \ldots, \ \gamma_m = f_m\,(\alpha).$$

D'après le théorème précédent ces séquences sont aussi périodiques et de périodes

$$p_1 = r'_1 p, \ \ldots, \ p_m = r'_m p \quad (r'_1, \ \ldots, \ r'_m \leqslant R).$$

Il est évident que ces périodes ne contiennent que des facteurs premiers non supérieurs à R. Considérons maintenant le vecteur

$$(\gamma_1, \ldots, \gamma_m);$$

c'est un vecteur périodique dont la période p_0 est le p.p.c.m. des périodes $p_1, \ldots, p_m$; donc, tous ses facteurs premiers seront non supérieurs à R. Enfin la séquence

$$\gamma = f_0 (\gamma_1, \ldots, \gamma_m)$$

sera, en vertu du théorème précédent, périodique et de période p':

$$p' = r'_0 p_0 \; (r'_0 \leqslant R).$$

Donc, p' ne contiendra aussi que des facteurs premiers non supérieurs à R. C.q.f.d.

Ces propositions vont nous servir à prouver le

Théorème 17. *Le système* (P^k_{db}, S) *ne possède pas de base finie.*

Démonstration. Supposons par absurde que le système (P^k_{db}, S) possède une base $f_1, \ldots, f_s$. Désignons par $r_1, \ldots, r_s$ les poids des f.d.b. $f_1, \ldots, f_s$. Supposons par ailleurs que

$$r = \max (r_1, \ldots, r_s)$$

et que p est un nombre premier tel que $p > r$. Considérons une fonction $f_\gamma (X)$ prenant une valeur identiquement égale à γ, où γ est une séquence périodique de période p, de la forme

$$\gamma = \underbrace{0 \ldots 01}_{p}\underbrace{0 \ldots 01}_{p} \ldots$$

La fonction $f_\gamma (X)$ ne peut être exprimée par une superposition au moyen des fonctions $f_1, \ldots, f_s$. En effet, si $f(X)$ était une superposition quelconque des fonctions $f_1, \ldots, f_s$, elle transformerait la séquence $0 = (0, 0, \ldots)$ (de période 1) en une séquence périodique dont la période ne contiendrait pas de facteurs premiers supérieurs à r. D'autre part, $f_\gamma (0) = \gamma$ est une séquence périodique de période $p > r$. Cette contradiction prouve le théorème.

L'opération R est essentielle puisque le système $(P^k_{\mathrm{db}}, S, R)$ possède une base finie et le système (P^k_{db}, S) n'en possède pas.

D'autre part, pour des raisons évidentes le système (P^k_{db}, R) ne possède pas de base finie, donc l'opération S est aussi essentielle.

CHAPITRE 4

FONCTIONS CALCULABLES

§ 19. Machines de Turing

De l'exposé précédent il résulte qu'une f.d.b. $f(x_1, \ldots, x_n)$ peut être définie par des équations canoniques

$$z(t) = F(X(t), Q(t-1)),$$
$$Q(t) = G(X(t), Q(t-1)),$$
$$Q(0) = 0.$$

Ces équations permettent de calculer la séquence de sortie d'après la séquence d'entrée des valeurs des variables $x_1, \ldots, x_n$. Bien plus, les valeurs de la séquence de sortie sont délivrées au fur et à mesure de l'arrivée des valeurs d'entrée. Ceci permet d'interpréter les équations ci-dessus comme la description du fonctionnement d'un convertisseur discret ou automate (fig. 23) à r états (r est le poids de la f.d.b.) qui travaille en temps discret en formant l'état de la mémoire et la valeur de sortie à l'instant t d'après l'état de la mémoire à l'instant $t-1$ et les valeurs d'entrée à l'instant t en vertu des équations canoniques. Ce dispositif ne cesse jamais de fonctionner contrairement aux automates réels (dispositifs automatiques).

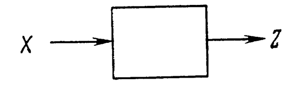


Fig. 23

Il est possible d'introduire une autre caractéristique fonctionnelle équivalente à l'initiale mais ayant un caractère fini. La fonction f étant déterminée, il s'ensuit que l'application f engendre une application de séquences finies de la forme $\{\alpha(1), \ldots, \alpha(i)\}$ dans des séquences finies $\{\gamma(1), \ldots, \gamma(i)\}$ ($i = 1, 2, \ldots$). Désignons cette application par $\varphi(x)$. La fonction φ peut être définie par les mêmes équations canoniques que f. Il est évident que la fonction $\varphi(x)$ restitue entièrement la fonction $f(x)$ (en ce sens, on a mentionné plus haut l'équivalence de φ et de f). Désignons par P^k_{db} la classe des fonctions $\varphi(x)$. La fonction $\varphi(x)$ peut être interprétée comme la description du fonctionnement d'un automate au sens suivant: aux instants $1, 2, \ldots, i$ sont appliqués à l'entrée les symboles $\alpha(1), \alpha(2), \ldots, \alpha(i)$ et à la sortie sont délivrés les symboles

$\gamma(1), \gamma(2), \ldots, \gamma(i)$; aux instants ultérieurs, aucun symbole n'apparaît sur l'entrée et l'automate cesse de fonctionner.

Dans ce cas, l'absence d'information à l'entrée traduit le fait qu'à la séquence d'entrée $\{\alpha(1), \ldots, \alpha(i)\}$ est associée une séquence infinie

$$\{\alpha(1), \ldots, \alpha(i), \Lambda, \ldots\},$$

où Λ est un symbole auxiliaire de l'alphabet d'entrée signifiant l'absence d'information. L'apparition du symbole Λ à l'entrée est la condition d'arrêt du dispositif, c'est-à-dire d'apparition à la sortie de la séquence

$$\{\gamma(1), \ldots, \gamma(i), \Lambda, \ldots\}.$$

Les dispositifs introduits qui fonctionnent sur des séquences d'entrée finies peuvent être assimilés à une « machine », constituée d'une bande illimitée à droite et d'un automate (voir fig. 24). La

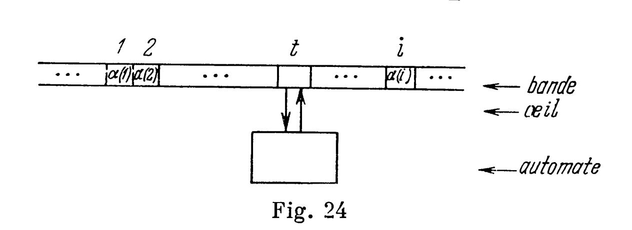


Fig. 24

bande illimitée est divisée en champs numérotés par les nombres 1, 2, . . .; les champs 1, 2, . . ., i contiennent les symboles $\alpha(1)$, $\alpha(2), \ldots, \alpha(i)$ de l'alphabet $(0, 1, \ldots, N-1)$. L'automate est équipé d'un œil et peut occuper l'un des états $\varkappa_1, \ldots, \varkappa_r$ (r est fini). A chaque instant t ($t = 1, 2, \ldots$) l'œil observe un champ de la bande (le premier champ pour $t = 1$) ; l'automate élabore un nouvel état en fonction du symbole observé et de l'état intérieur et imprime le nouveau symbole dans le champ observé (l'automate occupe l'état $\varkappa_1$ à l'instant initial $t = 0$). Ensuite, l'œil se déplace d'un champ à droite, etc. La machine s'arrête dès l'apparition du symbole Λ dans son champ visuel et les champs 1, 2, . . ., i affichent alors la séquence de sortie $\{\gamma(1), \gamma(2), \ldots, \gamma(i)\}$. Le travail de cette machine peut être décrit par un *programme*, c'est-à-dire un tableau spécial (voir tableau 18).

Dans ce tableau, les lignes sont numérotées par les symboles $\Lambda, 0, 1, \ldots, N-1$ et les colonnes, par les symboles $\varkappa_1, \ldots, \varkappa_r$. La ligne correspondant au symbole Λ est laissée vide. Dans la case qui se trouve à l'intersection de la ligne α ($\alpha \neq \Lambda$) et de la colonne j, on inscrit le triplet de symboles

$$\{F(\alpha, \varkappa_j), R, G(\alpha, \varkappa_j)\}.$$

Tableau 18

	$\varkappa_1$	...	$\varkappa_j$	...	$\varkappa_r$
Λ		...		...	
0		...		...	
...	...	...	...	...	...
α		...		...	
...	...	...	...	...	...
$N-1$		...		...	

Ce triplet s'appelle *instruction*. La machine exécute une instruction de la manière suivante : si l'œil observe le symbole α sur la bande et si à cet instant la machine se trouve dans l'état $\varkappa_j$, alors elle enregistre le symbole $F(\alpha, \varkappa_j)$ à la place de α, ensuite passe à l'état

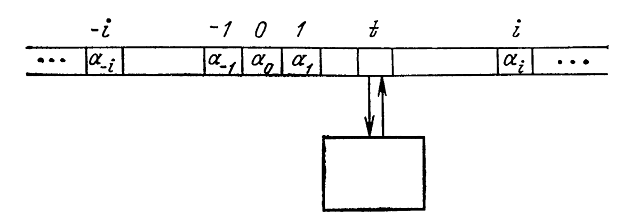


Fig. 25

$G(\alpha, \varkappa_j)$ et déplace l'œil d'un champ contigu à droite (R). Si le symbole observé est le symbole Λ, alors la case correspondante contient une instruction « vide » qui est considérée comme l'instruction d'arrêt de la machine. Il est évident que le programme de la machine est complètement défini par les équations canoniques pour $\varphi(x)$.

Ce type de machines peut être généralisé à une plus vaste classe de programmes et à un lien plus complexe entre l'automate et la bande.

Ainsi, la machine de la figure 25 (qui sera désignée par $\mathfrak{M}$) est constituée d'une bande illimitée dans les deux sens et d'un automate.

Les champs de la bande sont numérotés par les entiers . . ., $-i$, ..., -1, 0, 1, . . ., i, . . ., et contiennent les symboles de l'alphabet $\{0, 1, \ldots, k-1\}$ (dans la suite 0 jouera le rôle de symbole vide); l'automate est équipé d'un œil susceptible de réaliser l'une des fonctions suivantes: R, se déplacer d'un champ à droite, L, se déplacer d'un champ à gauche, S, poursuivre l'observation du même champ. Les symboles $\varkappa_1, \ldots, \varkappa_r$ désignent les états de l'automate. La machine $\mathfrak{M}$ fonctionne suivant un programme T (voir tableau 19). Une partie des cases peut rester vide, c'est-à-dire conte-

Tableau 19

	$\varkappa_1$	...	$\varkappa_j$	...	$\varkappa_r$
0		...		...	
...	...	...	...	...	...
a		...	$cD\varkappa$	...	
...	...	...	...	...	...
$k-1$		...		...	

nir des instructions vides, l'autre partie est remplie par des triplets de symboles constituant les instructions de la machine. Par exemple, dans la case située à l'intersection de la ligne a et de la colonne j (voir tableau 19) est inscrit le triplet $cD\varkappa$, c étant un symbole de l'alphabet $\{0, 1, \ldots, k-1\}$, D, un symbole de l'alphabet $\{R, L, S\}$, $\varkappa$ un des états $\{\varkappa_1, \ldots, \varkappa_r\}$.

Supposons que l'œil observe le symbole a à l'état $\varkappa_j$; alors:

a) si la case $(a, \varkappa_j)$ contient l'instruction $(cD\varkappa)$, la machine remplacera le symbole a de ce champ par c, passera à l'état $\varkappa$, effectuera le déplacement D et commencera à exécuter l'instruction suivante;

b) si la case $(a, \varkappa_j)$ renferme une instruction vide, la machine s'arrête.

A l'instant initial, l'œil observe le champ initial de la bande et la machine se trouve dans l'état $\varkappa_1$ (qui correspond à la colonne gauche du programme T).

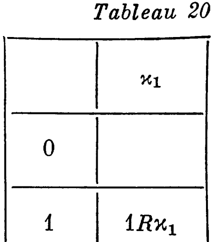
Donc, à partir de la situation initiale (état initial et champ initial) la machine commence à traiter l'enregistrement initial sur

la bande conformément au programme T. Deux cas peuvent se présenter: a) l'apparition d'une instruction vide provoque l'arrêt de la machine et l'on obtient l'enregistrement final sur la bande (l'état correspondant sera appelé *état final*), b) aucune instruction vide n'apparaît et la machine ne s'arrête pas.

Les machines introduites s'appellent *machines de Turing*.

Exemple 1. Supposons que $k = 2$. Considérons une machine qui dans un enregistrement quelconque trouve le premier zéro en se déplaçant à droite à partir d'un champ quelconque. Il est évident que cette machine peut être définie par le programme du tableau 20.

Tableau 20



	$\varkappa_1$
0	
1	$1R\varkappa_1$

En effet, trois cas sont possibles:

1) A l'instant initial l'œil observe le symbole 0 et la machine s'arrête immédiatement.

2) A l'instant initial l'œil observe le symbole 1 et à droite du champ initial l'enregistrement contient au moins un 0. La machine décale l'œil d'un champ à droite et s'arrête au-dessus du premier 0.

3) A l'instant initial l'œil détecte le symbole 1, et à droite du champ initial l'enregistrement ne contient que des 1. La machine déplacera la tête à droite sans s'arrêter.

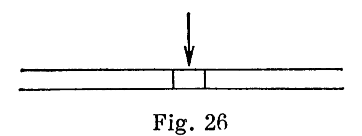


Fig. 26

Introduisons quelques notations et notions liées à l'enregistrement sur la bande. On représentera par des flèches la position de l'œil sur la bande à l'instant considéré (voir fig. 26).

Ceci peut encore être noté comme suit:

$$\ldots a_1\!\overset{\downarrow}{\,}a_2 \ldots$$

La flèche se rapporte au symbole a_1 situé immédiatement à gauche d'elle et signifie que l'œil observe le symbole a_1 à l'instant considéré.

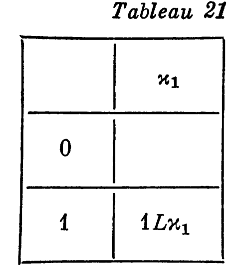
Si la bande n'est remplie que de 0, on dira quelquefois que nous avons affaire à une *bande vide*. On appellera aussi *champ vide* le champ affichant 0.

Enfin, l'ensemble des champs inspectés par l'œil entre l'instant initial et un instant t sera appelé *zone de travail de la bande à l'instant t*.

Dans la suite, nous aurons à construire des machines de Turing douées de propriétés spécifiques. Ces machines seront construites à partir de celles déjà connues. Pour ce faire nous introduisons le principe de dualité et deux types de compositions de machines.

Principe de dualité pour programmes (de machines). Soit T un programme quelconque. Désignons par T^* le programme déduit de T par substution de R à L et de L à R. Le programme T^* s'appelle *dual de T*.

Tableau 21



	$\varkappa_1$
0	
1	$1L\varkappa_1$

Exemple 2. Le programme T^* défini par le tableau 21 est visiblement dual de celui de l'exemple précédent.

Il est évident que $(T^*)^* = T$, c'est-à-dire que la notion de dualité est réciproque. Dans la suite, on appellera aussi *duales* les *machines* $\mathfrak{M}$ et $\mathfrak{M}^*$ correspondant aux programmes T et T^*. On voit immédiatement que les machines $\mathfrak{M}$ et $\mathfrak{M}^*$ fonctionnent dans un certain sens de manière symétrique, plus exactement: si à l'instant initial la bande affiche l'enregistrement

$$\ldots a_1^{\downarrow} a_2 \ldots \tag{1}$$

que la machine $\mathfrak{M}$ transforme à l'instant t en l'enregistrement

$$\ldots c_1 c_2 \ldots c_s^{\downarrow} \ldots, \tag{2}$$

alors la machine $\mathfrak{M}^*$ transforme à l'instant t l'enregistrement

$$\ldots a_2 a_1^{\downarrow} \ldots, \tag{3}$$

affiché à l'instant initial et symétrique de (1) par rapport à a_1 en l'enregistrement

$$\ldots c_s^{\downarrow} \ldots c_2 c_1 \ldots, \tag{4}$$

symétrique de (2) par rapport à c_1.

En vertu de cette remarque, la machine $\mathfrak{M}^*$ par exemple doit, en se déplaçant vers la gauche, chercher le premier zéro (voir exemple 2). On s'assure de ceci par une vérification immédiate.

Premier type de composition. Elle consiste à raccorder en série une machine à l'autre. Soient $\mathfrak{M}_0$ et $\mathfrak{M}_1$ deux machines de Turing sur un même alphabet d'entrée $\{0, 1, \ldots, k-1\}$, dont les ensembles d'états sont disjoints. Numérotons par les nombres $0, 1, 2, \ldots, l-1$ toutes les cases vides (les instructions) du programme T_0 de la machine $\mathfrak{M}_0$. Soit $p(x)$ un prédicat quelconque *) sur l'ensemble $\{0, 1, 2, \ldots, l-1\}$. Construisons une machine $\mathfrak{M}$ que nous appellerons *raccordement en série de la machine* $\mathfrak{M}_1$ *à la machine* $\mathfrak{M}_0$ (par rapport au prédicat $p(x)$). Pour cela dressons un nouveau tableau T (voir tableau 22) avec les tableaux T_0 et T_1 des machines

*) Un prédicat $p(x)$ sur $\{0, 1, \ldots, l-1\}$ est une fonction spéciale de P_l. Dans la suite, on aura affaire à des prédicats *bivalents* et *trivalents* qui prennent respectivement leurs valeurs sur les ensembles $\{0, 1\}$, $\{0, 1, 2\}$.

$\mathfrak{M}_0$ et $\mathfrak{M}_1$. La première moitié de T est confondue avec T_0 pour les cases de T_0 contenant une instruction non vide. Dans les cases η telles que $p(\eta) = 1$, le tableau T contient l'instruction $aS\varkappa'_1$ (où a est le numéro de la ligne contenant la case η, $\varkappa'_1$, l'état initial de la

Tableau 22

	$\varkappa_1, \ldots, \varkappa_j, \ldots, \varkappa_{r_0}$	$\varkappa'_1, \ldots, \varkappa'_m, \ldots, \varkappa'_{r_1}$
0		
...		
a		
...		
$k-1$		

machine $\mathfrak{M}_1$). Dans les cases η telles que $p(\eta) = 0$, le tableau T renferme aussi une instruction vide. La deuxième moitié du tableau T est entièrement confondue avec T_1.

La machine $\mathfrak{M}$ fonctionne de toute évidence comme suit: l'enregistrement initial de la bande est d'abord traité par la machine $\mathfrak{M}_0$ et si cette dernière cesse de fonctionner sur une instruction η telle que $p(\eta) = 1$, alors le contenu de la bande est traité par la machine $\mathfrak{M}_1$. Ceci étant, le champ initial de la machine $\mathfrak{M}_1$ sera celui sur lequel s'est arrêtée la machine $\mathfrak{M}_0$. Donc, la machine $\mathfrak{M}$ effectue dans un certain sens le travail en série des machines $\mathfrak{M}_0$ et $\mathfrak{M}_1$.

Deuxième type de composition ou itération de la machine. Soit $\mathfrak{M}_0$ une machine de Turing dont les cases vides du programme T_0 sont numérotées par les nombres $0, 1, 2, \ldots, l-1$. Supposons que $p(x)$ est un prédicat sur l'ensemble $\{0, 1, 2, \ldots, l-1\}$. Construisons une machine $\mathfrak{M}$ que nous appellerons *itération de la machine* $\mathfrak{M}_0$ *par rapport au prédicat* $p(x)$. Dressons le tableau T de la machine $\mathfrak{M}$ en nous servant de T_0. Le tableau T est confondu avec T_0 en dehors des cases vides de T_0. Dans les cases η de T_0 telles que $p(\eta) = 0$, le tableau T affiche l'instruction $aS\varkappa_1$ (où a est le numéro de la ligne contenant la case η, $\varkappa_1$, l'état initial de $\mathfrak{M}_0$). Dans les cases η de T_0 telles que $p(\eta) = 1$, le tableau T contient une instruction vide.

Il est immédiat de voir que la machine $\mathfrak{M}$ est dans un certain sens une itération de la machine $\mathfrak{M}$, c'est-à-dire que son travail équivaut à un travail répété de la machine $\mathfrak{M}_0$.

§ 20. Une méthode de construction de machines de Turing

On se propose de décrire une méthode de construction de machines de Turing faisant appel à la composition des machines et à un langage opératoriel spécial pour l'enregistrement des algorithmes. Ce langage a été proposé par A. Liapounov en 1953 (voir [8]). Comme il ne tient qu'un rôle auxiliaire nous ne donnerons pas sa définition logique rigoureuse, nous nous tiendrons à une brève description et examinerons quelques exemples.

1. Les objets de départ sont des opérateurs qui se subdivisent en trois groupes :

a) les opérateurs qui modifient la condition de la bande font varier les états de la machine et changent la position de l'œil. On les désignera par des majuscules latines A, B, . . . (parfois munies d'indices) ;

b) les opérateurs de contrôle des conditions logiques qui seront notés par les symboles $p^{\uparrow}$ ou $p_{\downarrow}^{\uparrow}$, affectés parfois d'indices ;

c) les opérateurs spéciaux représentés par les symboles $*$ et ω.

2. Les opérateurs sont utilisés à l'élaboration de circuits opératoriels d'après des règles déterminées. Un circuit opératoriel est une suite d'opérateurs dans laquelle chaque opérateur de contrôle des conditions logiques est relié par une flèche à un autre opérateur. Ainsi, l'expression

$$* A_0 p^{\uparrow} \overset{\downarrow}{A_1} \omega$$

représente un circuit opératoriel.

3. A tout circuit opératoriel correspond un algorithme caractérisant les modifications de la condition de la bande, des états de la machine et de la position de l'œil. Ces modifications sont réalisées à l'aide des règles suivantes.

a) Les opérateurs « travaillent » dans un ordre déterminé. Le premier à commencer est celui qui à l'instant donné est précédé du symbole $*$.

b) Supposons que nous ayons $*A$. Alors l'enregistrement de la bande, l'état de la machine et la position de l'œil à l'instant considéré sont modifiés par l'opérateur A en un autre enregistrement, un autre état et une autre position de l'œil. Ensuite la portion de circuit $*A$ se transforme en la portion $A*$, ce qui signifie que le circuit opératoriel se transforme aussi.

c) Supposons que nous ayons $*p^{\uparrow}$ (ou $*p_{\downarrow}^{\uparrow}$). Dans ce cas le prédicat p est calculé d'après l'enregistrement consigné sur la bande et l'état de la machine. Si $p = 1$, la portion $*p^{\uparrow}$ se transforme en $p^{\uparrow}*$, c'est-à-dire que nous passons à l'exécution de l'opérateur suivant ; si $p = 0$, la portion $*p^{\uparrow}$ se transforme en $p^{\overset{*}{\uparrow}}$, c'est-à-dire qu'est réalisé l'opérateur auquel mène la flèche.

Même chose pour $*p^{\uparrow}_{\downarrow}$, où p est un prédicat trivalent, et selon sa valeur on convient qu'a lieu l'une des transformations

$$* p^{\uparrow}_{\downarrow} \Rightarrow p^{\uparrow}_{\downarrow} *, \quad * p^{\uparrow}_{\downarrow} \Rightarrow p^{\overset{*}{\uparrow}}_{\downarrow}, \quad * p^{\uparrow}_{\downarrow} \Rightarrow p^{\uparrow}_{\underset{*}{\downarrow}}.$$

d) La combinaison $*\omega$ représente la fin des transformations ou du fonctionnement de la machine.

Exemple 3. Considérons le circuit opératoriel

$$* A_0 \overset{\ulcorner}{p} \overset{\downarrow}{A_1} \omega.$$

Supposons que les opérateurs A_0 et A_1 désignent les modifications de l'enregistrement de la bande, des états et de la position de l'œil, réalisées respectivement par les machines $\mathfrak{M}_0$ et $\mathfrak{M}_1$. Supposons que les états des machines $\mathfrak{M}_0$ et $\mathfrak{M}_1$ sont disjoints. Supposons enfin que $p = p(x)$ est un prédicat défini sur les numéros des champs (c'est-à-dire l'ensemble des couples $(a, \varkappa)$) où s'arrête la machine $\mathfrak{M}_0$. Alors le circuit opératoriel effectue la transformation suivante.

1. On a $*A_0$. La machine $\mathfrak{M}_0$ fonctionne jusqu'à l'arrêt (si celui-ci a lieu) au champ de numéro, disons, η. Le circuit se transforme en

$$A_0 * \overset{\ulcorner}{p} \overset{\downarrow}{A_1} \omega.$$

2. On a $*p$. On calcule le prédicat p et:

a) si $p = 1$, on obtient le circuit

$$A_0 \overset{\ulcorner}{p} * \overset{\downarrow}{A_1} \omega;$$

b) si $p = 0$, le circuit

$$A_0 \overset{\ulcorner}{p} \overset{\downarrow}{A_1} * \omega.$$

3. a) On a $*A_1$. La machine $\mathfrak{M}_1$ fonctionne. En cas d'arrêt de la machine $\mathfrak{M}_1$, on obtient le circuit

$$A_0 \overset{\ulcorner}{p} \overset{\downarrow}{A_1} * \omega.$$

b) On a $*\omega$. La transformation est finie.

La transformation réalisée par le circuit opératoriel donné est celle effectuée par le raccordement en série de la machine $\mathfrak{M}_1$ à $\mathfrak{M}_0$ par rapport au prédicat $p(x)$.

Exemple 4. Il est immédiat de voir que le circuit opératoriel

$$* \overset{\downarrow}{A_0} \overset{\urcorner}{p} \omega$$

peut être interprété comme le circuit qui définit la transformation obtenue par l'itération de la machine par rapport au prédicat p.

Passons maintenant à la description de la technique de programmation pour les machines de Turing, c'est-à-dire à la description d'une méthode d'élaboration d'un programme pour une machine de Turing réalisant une transformation donnée. Cette méthode sera simultanément illustrée par un exemple. Le processus de programmation comporte quatre étapes.

Première étape. Soit donnée une transformation de l'enregistrement et de la position de la tête. Supposons qu'il existe une machine de Turing réalisant cette transformation. On commence par des

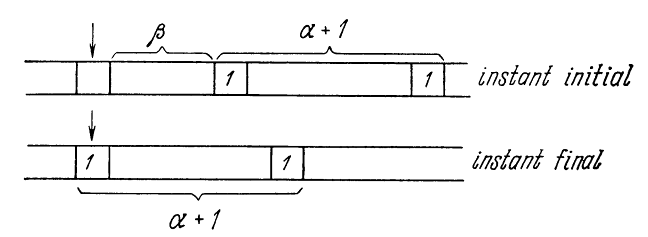


Fig. 27

raisonnements non formels à brosser un plan de réalisation de cette transformation. On s'arrange pour la décomposer en « transformations plus simples », c'est-à-dire des transformations pour lesquelles une machine de Turing a déjà été construite ou est facile à construire.

Exemple 5. Supposons que $k = 2$. Soit à construire une machine de Turing qui décale une zone de $\alpha + 1$ unités ($\alpha = 0, 1, \ldots$) de $\beta + 1$ champs à gauche ($\beta = 1, 2, \ldots$) (la bande est vierge en dehors de la zone). L'étendue $\beta + 1$ du décalage peut être définie par une disposition spéciale de l'œil à l'instant initial. La figure 27 représente les enregistrements et la position de l'œil aux instants initial et final. La transformation voulue peut être effectuée comme suit:

1) Marquons le champ initial en y remplaçant le symbole 0 par le symbole 1. Désignons par Φ ($0^{\downarrow} \rightarrow 1^{\downarrow}$) l'opérateur correspondant à cette transformation. Le contenu de la parenthèse indique que le symbole 0 a été remplacé par 1 et que l'œil reste en place.

2) Déplaçons-nous vers la droite (opérateur A_1) jusqu'à l'unité de gauche de la zone.

3) Désignons par a' et a'' respectivement le symbole observé à l'instant donné et le symbole situé immédiatement à droite de a'. Ici $a' = 1$. Voyons si l'unité, c'est-à-dire a', est la dernière unité de la zone. Calculons pour cela le prédicat $p(a', a'')$, où

$$p(a', a'') = \begin{cases} 0 \text{ pour } a'' = 0, \\ 1 \text{ pour } a'' = 1. \end{cases}$$

Pour $p = 1$

4) Effaçons a' (opérateur $0\,(a')$).

5) Retournons à gauche jusqu'à la plus proche unité (opérateur A_2).

6) Inscrivons une unité à droite de cette unité. Désignons par $\Phi\,(1^{\downarrow} \to 11^{\downarrow})$ la transformation correspondante.

7) Retournons (opérateur p_0, p_0 est un prédicat identiquement égal à 0) à A_1.

Pour $p = 0$

8) La zone est composée d'une seule unité. Effectuons $0\,(a')$, c'est-à-dire l'effacement de l'unité.

9) Retournons à gauche (opérateur A_2) jusqu'à la plus proche unité.

10) Déplaçons-nous à gauche (opérateur L) le long de la zone des unités et arrêtons-nous à l'unité de gauche.

Deuxième étape. Passage du plan établi au circuit opératoriel.

Suite de l'exemple 5. Dans ce cas le circuit opératoriel est de la forme

$$* \Phi\,(0^{\downarrow} \to 1^{\downarrow}) \uparrow A_1 p\,(a'a'') \overline{\,|\, 0\,(a')\, A_2 \Phi\,(1^{\downarrow} \to 11^{\downarrow})\, p_0 \,|\,} \downarrow 0\,(a')\, A_2 L \omega .$$

Remarque. Pendant l'élaboration du circuit opératoriel il faut veiller à ce que l'œil observe le même champ après la réalisation d'un opérateur et avant celle du suivant. Parfois on est amené à introduire pour cela des opérateurs concordants supplémentaires.

Dans l'exemple considéré, cette concordance est obtenue de façon naturelle sauf pour l'opérateur $p\,(a'a'')$, à propos duquel on admet qu'après sa réalisation l'œil observe le symbole a'.

Troisième étape. Passage du circuit opératoriel au programme de la machine.

On compose d'abord les programmes pour chaque opérateur du circuit opératoriel (sauf pour les opérateurs $*$, p_0, ω).

Tableau 23

$\Phi\,(0^{\downarrow} \to 1^{\downarrow})$	$\varkappa_1$
0	$1S\varkappa_1$
1	

A_1	$\varkappa_2$	$\varkappa_3$
0		$0R\varkappa_3$
1	$1R\varkappa_3$	

$p\,(a'a'')$	$\varkappa_4$	$\varkappa_5$	$\varkappa_6$	$\varkappa_7$
0		$0L\varkappa_7$		
1	$1R\varkappa_5$	$1L\varkappa_6$		

$0\,(a')$	$\varkappa_8$
0	
1	$0S\varkappa_8$

A_2	$\varkappa_9$
0	$0L\varkappa_9$
1	

$\Phi\,(1^{\downarrow} \to 11^{\downarrow})$	$\varkappa_{10}$	$\varkappa_{11}$
0	$1S\varkappa_{11}$	
1	$1R\varkappa_{10}$	

L	$\varkappa'_{14}$	$\varkappa''_{15}$
0	$0R\varkappa''$	
1	$1L\varkappa'$	

Suite de l'exemple 5. Nous avons les programmes suivants pour les opérateurs figurant dans le circuit (si des opérateurs interviennent plusieurs fois dans un circuit, le programme n'est composé que pour l'un d'eux ; voir tableau 23). On choisira les états de telle sorte que les programmes des divers opérateurs ne se coupent pas.

D'autre part, un circuit opératoriel peut être considéré comme un circuit déterminant la composition des programmes (de machines). Pour cela on « brisera » toutes les flèches (sauf celles de type $\downarrow$) et on admettra que chaque extrémité de flèche conduit à son état d'arrêt.

Dans l'exemple 5 on obtient le circuit suivant :

$$* \Phi (0^{\downarrow} \rightarrow 1^{\downarrow}) A_1 p (a'a'')^{\uparrow} 0 (a') A_2 \Phi (1^{\downarrow} \rightarrow 11^{\downarrow}) p_{0\downarrow} 0 (a') A_2 L \omega.$$

Ensuite, en prenant la portion de circuit

$$A_1 p (a'a'')^{\uparrow} 0 (a') A_2 \Phi (1^{\downarrow} \rightarrow 11^{\downarrow}),$$

on construit le programme qui est un raccordement en série des programmes correspondants. Puis on passe à la portion

$$\underline{\uparrow A_1 p (a'a'')^{\uparrow} 0 (a') A_2 \Phi (1^{\downarrow} \rightarrow 11^{\downarrow}) p_0 |}$$

et on construit le programme en appliquant l'opération d'itération au programme précédent par rapport au prédicat p_0. Considérons la portion

$$* \Phi (0^{\downarrow} \rightarrow 1^{\downarrow}) \underline{\uparrow A_1 p (a'a'') \overline{| 0 (a') A_2 \Phi (1^{\downarrow} \rightarrow 11^{\downarrow}) p_0} |} \downarrow$$

et construisons le programme correspondant en raccordant en série le programme précédent au programme de $\Phi (0^{\downarrow} \rightarrow 1^{\downarrow})$. Enfin, le circuit

$$* \Phi (0^{\downarrow} \rightarrow 1^{\downarrow}) \underline{\uparrow A_1 p (a'a'') \overline{| 0 (a') A_2 \Phi (1^{\downarrow} \rightarrow 11^{\downarrow}) p_0 |}} \downarrow 0 (a') A_2 L \omega$$

nous conduit à un programme qui est un raccordement en série du programme construit et des programmes pour $0 (a')$, A_2 et L. Le tableau 24 représente le programme sous sa forme définitive.

Remarque. Lors du raccordement en série des programmes correspondant à deux opérateurs consécutifs B' et B'' formant la portion $B'B''$ du circuit opératoriel, le prédicat p est supposé égal à un identiquement si l'opérateur B' est un opérateur qui modifie la condition de la bande, l'état de la machine et la position de l'œil. Si B' est l'opérateur de contrôle de la condition logique, le prédicat p est choisi en fonction de cette condition.

Quatrième étape. Simplification du programme. Les programmes construits par cette méthode donnent lieu parfois à d'importantes

Tableau 24

	$\Phi(0^{\downarrow} \to 1^{\downarrow})$	A_1		$p(a'a'')$				$0(a')$
	$\varkappa_1$	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$	$\varkappa_5$	$\varkappa_6$	$\varkappa_7$	$\varkappa_8$
0	$1S\varkappa_1$		$0R\varkappa_3$		$0L\varkappa_7$	$0S\varkappa_8$	$0S\varkappa_{12}$	$0S\varkappa_9$
1	$1S\varkappa_2$	$1R\varkappa_3$	$1S\varkappa_4$	$1R\varkappa_5$	$1L\varkappa_6$	$1S\varkappa_8$	$1S\varkappa_{12}$	$0S\varkappa_8$

	A_2	$\Phi(1^{\downarrow} \to 11^{\downarrow})$		$0(a')$	A_2	L	ω
	$\varkappa_9$	$\varkappa_{10}$	$\varkappa_{11}$	$\varkappa_{12}$	$\varkappa_{13}$	$\varkappa_{14}$	$\varkappa_{15}$
0	$0L\varkappa_9$	$1S\varkappa_{11}$	$0S\varkappa_2$	$0S\varkappa_{13}$	$0L\varkappa_{13}$	$0R\varkappa_{15}$	
1	$1S\varkappa_{10}$	$1R\varkappa_{10}$	$1S\varkappa_2$	$0S\varkappa_{12}$	$1S\varkappa_{14}$	$1L\varkappa_{14}$	

simplifications portant sur le nombre d'états. Nous allons formuler ici quelques règles de simplification.

Supposons qu'un programme présente deux états $\varkappa'$ et $\varkappa''$ tels que les colonnes correspondantes possèdent des cases vides auxquelles la machine n'accèdera jamais. Supposons que dans chaque ligne il existe une case vide de ce type (ou tout au moins dans l'une des colonnes indiquées). Il est immédiat de voir que l'on peut alors identifier les états $\varkappa'$ et $\varkappa''$.

Un autre type de simplification est lié aux opérateurs de contrôle des conditions logiques. Illustrons-le sur notre exemple.

Tableau 25

	$\varkappa_1$
0	$1S\varkappa_2$
1	

Suite de l'exemple 5. Les états $\varkappa_6$ et $\varkappa_7$ sont liés aux instructions qui réalisent uniquement les passages à d'autres états. On peut les éliminer par correction de l'instruction pour $\varkappa_5$.

Il existe encore un procédé de simplification lié aux instructions contenant le symbole du mouvement S. Supposons que la case $(c, \varkappa)$ renferme l'instruction $c'D\varkappa'$. Si on ne peut passer directement à cette instruction qu'à partir des instructions $cS\varkappa$ et si de plus elle ne fonctionne pas à l'instant initial, on peut alors retirer l'instruction $c'D\varkappa'$ de la case $(c, \varkappa)$ et remplacer toutes les instructions $cS\varkappa$ par $c'D\varkappa'$.

Suite de l'exemple 5. Dans le programme du tableau 24, on ne peut directement accéder à l'instruction $1S\varkappa_2$ consignée dans la case $(1, \varkappa_1)$ qu'à partir de l'instruction $1S\varkappa_1$ de la même colonne.

Donc, on peut remplacer la première colonne du programme par la suivante (voir tableau 25). On peut ensuite identifier les états $\varkappa_1$ et $\varkappa_2$ puisque les cases $(1, \varkappa_1)$ et $(0, \varkappa_2)$ contiennent des instructions vides et la machine n'y accédera jamais.

On peut effectuer la même transformation avec les colonnes correspondant à $\varkappa_3$ et $\varkappa_4$: retirer l'instruction $1R\varkappa_5$ et identifier les

Tableau 26

	$\varkappa_1$	$\varkappa_3$	$\varkappa_5$	$\varkappa_8$	$\varkappa_9$	$\varkappa_{10}$	$\varkappa_{12}$	$\varkappa_{13}$	$\varkappa_{14}$	$\varkappa_{15}$
0	$1S\varkappa_1$	$0R\varkappa_3$	$0L\varkappa_{12}$	$0S\varkappa_9$	$0L\varkappa_9$	$1S\varkappa_1$	$0S\varkappa_{13}$	$0L\varkappa_{13}$	$0R\varkappa_{15}$	
1	$1R\varkappa_3$	$1R\varkappa_5$	$1L\varkappa_8$	$0S\varkappa_8$	$1S\varkappa_{10}$	$1R\varkappa_{10}$	$0S\varkappa_{12}$	$1S\varkappa_{14}$	$1L\varkappa_{14}$	

états $\varkappa_3$ et $\varkappa_4$. Enfin on peut retirer l'instruction $0S\varkappa_2$ de la colonne $\varkappa_{11}$, car elle n'est jamais mise en œuvre et en faire de même pour $1S\varkappa_2$ en procédant aux modifications indispensables dans la colonne correspondant à $\varkappa_{10}$.

Nous obtenons un programme (voir tableau 26) constitué de 10 états.

§ 21. Les codes de machine et leurs conversions

Dans la suite nous n'étudierons que les machines dont l'alphabet d'entrée est composé de deux symboles.

Le fonctionnement de la machine de Turing dépend de l'enregistrement initial de la bande. Dans la suite on utilisera souvent des formes spéciales de ces enregistrements, appelés *codes de machine.* On distinguera deux types de codes: les codes principaux et les codes auxiliaires.

Les *codes principaux* sont soit de la forme:

$$\ldots 0\underbrace{1\ldots 1}_{\alpha+1}0\ldots$$

i.e. une zone de $\alpha+1$ unités; soit de la forme

$$\ldots 0\underbrace{1\ldots 1}_{\alpha_1+1}0\underbrace{1\ldots 1}_{\alpha_2+1}0\ldots 0\underbrace{1\ldots 1}_{\alpha_s+1}0\ldots$$

i.e. s zones respectivement de α_1+1, α_2+1, . . ., α_s+1 unités, séparées par un zéro.

Les codes principaux sont utilisés pour définir les nombres α et les combinaisons de nombres α_1, α_2, . . ., α_s de $E^{\aleph_0}$ (c'est-à-dire

de l'ensemble des entiers naturels et du zéro). Le zéro est codé par un enregistrement qui contient exactement une unité.

De nombreux problèmes portent sur les codes principaux. Nous allons examiner l'un d'eux qui consiste à trouver l'unité de gauche dans un code principal. De façon plus exacte, on demande de construire une machine qui laisse invariant le code principal quelle que soit la position initiale de l'œil et s'arrête sur l'unité de gauche de ce code.

Donnons la solution détaillée de ce problème.

Première étape. Plan de travail de la machine cherchée. Supposons que l'enregistrement initial est de la forme $\ldots a_0 \overset{\downarrow}{a_1} a_2 \ldots$

1) Voyons si le champ initial n'est pas vide, c'est-à-dire vérifions la condition $p\,(a_1 \neq 0)$.

2) Supposons que $p\,(a_1 \neq 0) = 1$, c'est-à-dire qu'à l'instant initial l'œil observe le symbole 1. On cherche alors l'extrémité gauche (c'est-à-dire l'unité de gauche) du code principal (opérateur A_1) et on s'arrête.

3) Supposons que $p\,(a_1 \neq 0) = 0$. On vérifie alors si le champ contigu à gauche de a_1 est vide, c'est-à-dire qu'on vérifie la condition $p\,(a_0 = 0)$.

4) Si $p\,(a_0 = 0) = 0$, on retourne à la réalisation de l'opérateur A_1.

5) Si $p\,(a_0 = 0) = 1$, c'est-à-dire que $a_0 = 0$, alors on remplace les symboles a_0 et a_1 par deux unités et on s'arrête sur celle de gauche (opérateur $\Phi\,(\overset{\downarrow}{a_0} a_1 \to \overset{\downarrow}{1}1)$). Donc, en dehors de la zone d'unités on a construit un segment dont les extrémités sont des unités. Sa longueur initiale est égale à deux, mais dans la suite nous l'étendrons en décalant l'unité de gauche à gauche et l'unité de droite à droite.

6) Voyons s'il est possible de décaler l'unité de gauche du segment d'un champ à gauche moyennant la vérification de la condition p_g, où

$$p_g = \begin{cases} 1 \text{ si le champ contigu à gauche du segment est vide,} \\ 0 \text{ dans le cas contraire.} \end{cases}$$

Supposons que $p_g = 1$. Alors

7) Décalons l'unité de gauche d'un champ à gauche et déplaçons-nous ensuite à droite jusqu'à l'unité de droite du segment (opérateur A_2).

8) Voyons s'il est possible de décaler la première unité de droite du segment d'un champ à droite moyennant la vérification de la condition p_d, où

$$p_d = \begin{cases} 1 \text{ si le champ contigu à droite du segment est vide,} \\ 0 \text{ dans le cas contraire.} \end{cases}$$

Si $p_d = 1$, alors

9) Décalons l'unité de droite d'un champ à droite, puis déplaçons-nous à gauche jusqu'à l'unité de gauche (opérateur A_3) et retournons à 6).

Si $p_g = 0$, alors

10) l'unité de gauche touche la zone d'unités. Déplaçons-nous à droite et effaçons les deux unités du segment. Puis retournons jusqu'à l'extrémité droite du segment (opérateur A_4) et passons à A_1.

Si $p_d = 0$, alors

Tableau 27

$p\ (a_1 \neq 0)$	$\varkappa_1$
0	
1	

A_1	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$
0	$0L\varkappa_3$	$0R\varkappa_4$	$0R\varkappa_4$
1	$1L\varkappa_2$	$1L\varkappa_2$	

$p\ (a_0 = 0)$	$\varkappa_5$	$\varkappa_6$
0	$0L\varkappa_6$	
1		

Φ	$\varkappa_7$	$\varkappa_8$
0	$1R\varkappa_8$	$1L\varkappa_8$
1		

p_g	$\varkappa_9$	$\varkappa_{10}$
0		
1	$1L\varkappa_{10}$	

A_2	$\varkappa_{11}$	$\varkappa_{12}$
0	$1R\varkappa_{11}$	$0R\varkappa_{12}$
1	$0R\varkappa_{12}$	

p_d	$\varkappa_{13}$	$\varkappa_{14}$
0		
1	$1R\varkappa_{14}$	

A_3	$\varkappa_{15}$	$\varkappa_{16}$
0	$1L\varkappa_{15}$	$0L\varkappa_{16}$
1	$0L\varkappa_{16}$	

A_4	$\varkappa_{17}$	$\varkappa_{18}$	$\varkappa_{19}$	$\varkappa_{20}$
0			$0R\varkappa_{19}$	$0L\varkappa_{20}$
1	$1R\varkappa_{18}$	$0R\varkappa_{19}$	$0L\varkappa_{20}$	

A_5	$\varkappa_{21}$	$\varkappa_{22}$	$\varkappa_{23}$	$\varkappa_{24}$
0			$0L\varkappa_{23}$	$0R\varkappa_{24}$
1	$1L\varkappa_{22}$	$0L\varkappa_{23}$	$0R\varkappa_{24}$	

11) L'unité de droite touche la zone d'unités. Déplaçons-nous vers la gauche et effaçons les deux unités du segment. Revenons ensuite jusqu'à l'extrémité gauche de la zone (opérateur A_5) et arrêtons-nous.

Deuxième étape. Ecriture du circuit opératoriel. Il est de la forme

$$* \; p\,(a_1 \neq 0)\, A_1 \omega p\,(a_0 = 0)\, \Phi\,(a_0^{\downarrow} a_1 \rightarrow 1^{\downarrow} 1)\, p_g A_2 p_d A_3 p_0 A_4 p_0 A_5 \omega$$

Troisième étape. Rédaction des programmes pour les divers opérateurs (voir tableau 27). Les programmes pour p_d, A_3 et A_5 sont duaux respectivement de p_g, A_2 et A_4.

Rédigeons le programme du problème posé (voir tableau 28) en nous servant du circuit opératoriel et des programmes pour les opérateurs.

Tableau 28

	$p\,(a_1 \neq 0)$	A_1			$p\,(a_0 = 0)$		$\Phi\,(a_0^{\downarrow} a_1 \rightarrow 1^{\downarrow} 1)$	
	$\varkappa_1$	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$	$\varkappa_5$	$\varkappa_6$	$\varkappa_7$	$\varkappa_8$
0	$0S\varkappa_5$	$0L\varkappa_3$	$0R\varkappa_4$	$0R\varkappa_4$	$0L\varkappa_6$	$0S\varkappa_7$	$1R\varkappa_8$	$1L\varkappa_8$
1	$1S\varkappa_2$	$1L\varkappa_2$	$1L\varkappa_2$			$1S\varkappa_2$		$1S\varkappa_9$

	p_g		A_2		p_d		A_3	
	$\varkappa_9$	$\varkappa_{10}$	$\varkappa_{11}$	$\varkappa_{12}$	$\varkappa_{13}$	$\varkappa_{14}$	$\varkappa_{15}$	$\varkappa_{16}$
0		$0S\varkappa_{11}$	$1R\varkappa_{11}$	$0R\varkappa_{12}$		$0S\varkappa_{15}$	$1L\varkappa_{15}$	$0L\varkappa_{16}$
1	$1L\varkappa_{10}$	$1S\varkappa_{17}$	$0R\varkappa_{12}$	$1S\varkappa_{13}$	$1R\varkappa_{14}$	$1S\varkappa_{21}$	$0L\varkappa_{16}$	$1S\varkappa_9$

	A_4				A_5			
	$\varkappa_{17}$	$\varkappa_{18}$	$\varkappa_{19}$	$\varkappa_{20}$	$\varkappa_{21}$	$\varkappa_{22}$	$\varkappa_{23}$	$\varkappa_{24}$
0			$0R\varkappa_{19}$	$0L\varkappa_{20}$			$0L\varkappa_{23}$	$0R\varkappa_{24}$
1	$1R\varkappa_{18}$	$0R\varkappa_{19}$	$0L\varkappa_{20}$	$1S\varkappa_2$	$1L\varkappa_{22}$	$0L\varkappa_{23}$	$0R\varkappa_{24}$	

Quatrième étape. Simplifions le programme. Nous pouvons retirer les instructions situées dans les colonnes $\varkappa_5$, $\varkappa_7$, $\varkappa_9$, $\varkappa_{13}$, $\varkappa_{17}$, $\varkappa_{21}$ car nous pouvons y accéder directement à partir de $0S\varkappa_5$, $0S\varkappa_7$, $1S\varkappa_9$,

$1S\varkappa_{13}$, $1S\varkappa_{17}$, $1S\varkappa_{21}$. Nous pouvons ensuite éliminer les états $\varkappa_5$, $\varkappa_7$, $\varkappa_9$, $\varkappa_{13}$, $\varkappa_{17}$, $\varkappa_{21}$. Nous obtenons un programme à 18 états (voir tableau 29) *).

Tableau 29

	$\varkappa_1$	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$	$\varkappa_6$	$\varkappa_8$	$\varkappa_{10}$	$\varkappa_{11}$	$\varkappa_{12}$
0	$0L\varkappa_6$	$0L\varkappa_3$	$0R\varkappa_4$	$0R\varkappa_4$	$1R\varkappa_8$	$1L\varkappa_8$	$0S\varkappa_{11}$	$1R\varkappa_{11}$	$0R\varkappa_{12}$
1	$1S\varkappa_2$	$1L\varkappa_2$	$1L\varkappa_2$		$1S\varkappa_2$	$1L\varkappa_{10}$	$1R\varkappa_{18}$	$0R\varkappa_{12}$	$1R\varkappa_{14}$

	$\varkappa_{14}$	$\varkappa_{15}$	$\varkappa_{16}$	$\varkappa_{18}$	$\varkappa_{19}$	$\varkappa_{20}$	$\varkappa_{22}$	$\varkappa_{23}$	$\varkappa_{24}$
0	$0S\varkappa_{15}$	$1L\varkappa_{15}$	$0L\varkappa_{16}$		$0R\varkappa_{19}$	$0L\varkappa_{20}$		$0L\varkappa_{23}$	$0R\varkappa_{24}$
1	$1L\varkappa_{22}$	$0L\varkappa_{16}$	$1S\varkappa_8$	$0R\varkappa_{19}$	$0L\varkappa_{20}$	$1S\varkappa_2$	$0L\varkappa_{23}$	$0R\varkappa_{24}$	

Dans la suite on ne composera en principe les programmes que jusqu'à la deuxième étape, c'est-à-dire jusqu'à l'élaboration des circuits opératoriels, car la partie restante ne présente pas de difficultés.

Le principe de dualité permet de rédiger sans peine un programme de localisation de l'unité de droite du code principal. En compliquant un peu les choses on peut composer un programme pour la localisation de l'unité de droite ou de gauche de la *i*-ème zone du code principal.

Passons maintenant à la description des codes auxiliaires. Nous distinguerons trois types de *codes auxiliaires.*

a) Le *code l-multiple* est défini pour une combinaison quelconque $\alpha_1, \alpha_2, \ldots, \alpha_s$ des nombres de $E^{\aleph_0}$ de la manière suivante:

$$\ldots 0\underbrace{1\ldots 1}_{l(\alpha_1+1)}U\underbrace{1\ldots 1}_{l(\alpha_2+1)}U\ldots U\underbrace{1\ldots 1}_{l(\alpha_s+1)}0\ldots,$$

où U est un mot tampon de longueur l et $U = 0U'$, c'est-à-dire que U commence par un zéro;

b) le *code réticulé* est défini pour une combinaison quelconque $\alpha_1, \alpha_2, \ldots, \alpha_s$ de $E^{\aleph_0}$. C'est un enregistrement que l'on peut décomposer à l'aide de s réseaux (s suites de champs de la bande) de période s en zones d'unités, plus exactement: le premier réseau contient une zone de $\alpha_1 + 1$ unités, le deuxième, de $\alpha_2 + 1$ unités, etc.,

*) On peut identifier $\varkappa_4$ et $\varkappa_{24}$ dans ce tableau.

le s-ième, de $\alpha_s + 1$ unités, les origines de ces zones étant compatibles, c'est-à-dire se suivent sur la bande à la manière des numéros des réseaux (voir fig. 28 ; pour plus de suggestion chaque réseau a été représenté séparément) ;

c) Le *code quasi principal* est défini pour un code principal quelconque $b_1 b_2 \ldots b_\nu$, où $b_1 = b_\nu = 1$, sous forme de l'écriture suivante :

$$b_1 U_1 b_2 U_2 \ldots U_{\nu-1} b_\nu,$$

où

$$|U_1| = \ldots = |U_{\nu-1}| = l - 1 \quad (l \geqslant 2).$$

Donc, dans le code quasi principal, un réseau de pas l contient le

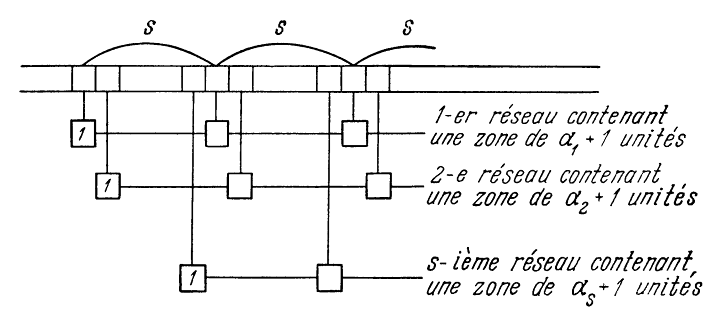


Fig. 28

code principal, les intervalles intérieurs sont occupés par les mots $U_1, U_2, \ldots, U_{\nu-1}$, la partie restante de la bande est vide.

Les codes de machine et leurs conversions sont justiciables par de nombreux lemmes.

Lemme 1 (de simulation sur un réseau). *Soient $\mathfrak{M}$ une machine de Turing de programme T, l un entier $\geqslant 2$. On peut alors construire une machine $\widetilde{\mathfrak{M}}$ qui sur un réseau de pas l fonctionne comme la machine $\mathfrak{M}$ sur la bande tout entière.*

Démonstration. Construisons un tableau $\widetilde{T}$ à partir du tableau T (voir tableau 30) en ajoutant à chaque état $\varkappa_j$ les états auxiliaires $\varkappa_j^1, \ldots, \varkappa_j^{l-1}, \varkappa_j^l, \ldots, \varkappa_j^{2l-2}$, destinés pour le passage des champs situés en dehors du réseau et la mémorisation du caractère du déplacement (voir tableau 31), où

$$\varkappa_j^D = \begin{cases} \varkappa_j \text{ pour } D = S, \\ \varkappa_j^1 \text{ pour } D = R, \\ \varkappa_j^l \text{ pour } D = L. \end{cases}$$

Tableau 30

	$\varkappa_1$	...	$\varkappa_i$	...	$\varkappa_j$	...	$\varkappa_r$
0							
...							
a			$cD\varkappa_i$				
...							
$k-1$							

Sur le tableau $\tilde{T}$, on voit que la machine $\tilde{\mathfrak{M}}$ lorsqu'elle se trouve dans un champ du réseau et détecte le symbole a dans l'état $\varkappa_i$, le remplace par c comme la machine $\mathfrak{M}$, effectue le même déplacement D et passe à l'état $\varkappa_j^D$. L'état $\varkappa_j^D$ dépend du caractère du déplacement : ce sera $\varkappa_j$ pour $D = S$, $\varkappa_j^1$ pour $D = R$ et $\varkappa_j^l$ pour $D = L$. Dans les deux derniers cas, l'œil quitte le réseau et la machine passe par les états $\varkappa_j^1, \varkappa_j^2, \ldots, \varkappa_j^{l-1}$ si l'œil se déplace vers la droite, et par les états $\varkappa_j^l, \varkappa_j^{l+1}, \ldots, \varkappa_j^{2l-2}$ s'il se déplace vers la gauche. Ensuite l'œil arrive au réseau d'état $\varkappa_j$ en se déplaçant d'un champ sur ce réseau. Donc, dans le cas des déplacements R et L, la machine $\tilde{\mathfrak{M}}$ fait exactement la même chose que la machine $\mathfrak{M}$ sur la bande tout entière, mais en l pas.

Corollaires. 1) *Si* $C_j(a) \equiv a$ *pour* $a = 0, \ldots, k-1$ *et pour tout* j, *alors la machine* $\tilde{\mathfrak{M}}$ *ne modifie pas l'enregistrement de la bande en dehors du réseau.*

2) *Si* $C_j(a) \equiv 0$ *pour* $a = 0, \ldots, k-1$ *et pour tout* j, *alors la machine* $\tilde{\mathfrak{M}}$ *efface la zone de travail.*

3) *Si* $C_j(a) \equiv 1$ *pour* $a = 0, \ldots, k-1$ *et pour tout* j *alors la machine affiche* 1 *dans la zone de travail et marque par là même les champs situés en dehors du réseau qui ont été inspectés par l'œil.*

4) *Des situations mixtes sont possibles, par exemple* $C_1(a) = C_{2l-2}(a) = 1$ *pour* $a = 0, \ldots, k-1$ *et* $C_j(a) = a$ *dans les autres cas. Ici la machine* $\tilde{\mathfrak{M}}$ *affiche* 1 *dans les limites de la zone de travail sur le réseau voisin qui n'est autre que le réseau initial décalé*

	...	$\varkappa_i$	...	$\varkappa_j^1$	$\varkappa_j^2$	...	$\varkappa_j^{l-1}$	$\varkappa_j^l$	...	$\varkappa_j^{2l-2}$	...
0	...	...	...	$C_1(0)\,R\varkappa_j^2$	$C_2(0)\,R\varkappa_j^3$	...	$C_{l-1}(0)\,R\varkappa_j$	$C_l(0)\,L\varkappa_j^{l+1}$	...	$C_{2l-2}(0)\,L\varkappa_j$	...
...	...	...	...	...	...	...	...	...	...	...	...
a	...	$cD\varkappa_j^D$	...	$C_1(a)\,R\varkappa_j^2$	$C_2(a)\,R\varkappa_j^3$	...	$C_{l-1}(a)\,R\varkappa_j$	$C_l(a)\,L\varkappa_j^{l+1}$	...	$C_{2l-2}(a)\,L\varkappa_j$	...
...	...	...	...	...	...	...	...	...	...	...	...
$k-1$	...	...	...	$C_1(k-1)\,R\varkappa_j^2$	$C_2(k-1)\,R\varkappa_j^3$	...	$C_{l-1}(k-1)\,R\varkappa_j$	$C_l(k-1)\,L\varkappa_j^{l+1}$	...	$C_{2l-2}(k-1)\,L\varkappa_j$	...

d'une unité vers la droite, et ne modifie pas le contenu de la bande en dehors de ces réseaux.

Introduisons la notation suivante. A étant un opérateur modifiant la condition de la bande, on désignera par $\tilde{A}$ l'opérateur simulant A sur un réseau (son comportement en dehors du réseau sera expressément spécifié).

Lemme 2 (de conversion du code principal en code l-multiple). *Soit l un entier naturel ($\geqslant 2$). On peut alors construire une machine de Turing convertissant le code principal en un code l-multiple avec un mot tampon U donné, où $|U| = l$ et $U = 0U'$.*

Démonstration. *Première étape.* La machine cherchée élaborera pas à pas le code l-multiple sur la bande à droite du code principal. Pour cela elle parcourt de gauche à droite les symboles du code principal, les remplace un par un par zéro et à droite du code principal inscrit une zone de l unités ou le mot U selon que le symbole remplacé est 1 ou 0. Ce processus fait apparaître sur la bande un mot dans lequel l'intervalle de deux unités consécutives ne peut contenir plus de l zéros. Pour de tels mots on peut (comme pour les codes principaux) construire une machine qui en localisera l'extrémité gauche (resp. droite). Désignons les déplacements de l'œil vers la gauche et vers la droite respectivement par L et R.

On peut donc décrire le fonctionnement de la machine de façon plus précise :

1) on accède à l'extrémité droite du code principal (opérateur R) ;

2) on se déplace de deux champs vers la droite et on inscrit la zone de l unités. Désignons cet opérateur par $\Phi\,(a^{\downarrow} \rightarrow a\,00\underbrace{1\,\ldots\,1}_{l}{}^{\downarrow})$;

3) on retourne à l'extrémité gauche du code principal (opérateur L) ;

4) on lit les trois premiers symboles $a'a''a'''$ à partir de l'extrémité gauche du code principal (et ensuite dans la partie restante du code principal) et on calcule le prédicat trivalent $p\,(a', a'', a''')$.

Si $a'a''a''' = 11a'''$, on se trouve en régime I et on passe à l'opérateur indiqué par la flèche $\uparrow$ ($p^{\uparrow}$).

Si $a'a''a''' = 101$, on se trouve en régime II et on passe à l'opérateur qui suit p.

Si $a'a''a''' = 100$, on se trouve en régime III et on passe à l'opérateur indiqué par la flèche $\downarrow$ ($p_{\downarrow}$).

Régime I.

5) On efface le symbole a' (opérateur $0\,(a')$) ;

6) On se rend à l'extrémité droite du mot (opérateur R) ;

7) On place la zone de l unités immédiatement à droite du mot (opérateur $\Phi\,(a^{\downarrow} \rightarrow a\underbrace{1\,\ldots\,1}_{l}{}^{\downarrow})$) ;

8) On retourne à l'extrémité gauche du mot (opérateur L) et on passe à 4).

Régime II (le premier symbole a' est le dernier de la zone des unités du code principal, mais il existe au moins une zone d'unités non traitée dans le code principal).

9) On efface le symbole a' $(0\,(a'))$;

10) On se rend vers la fin du mot (opérateur R);

11) On place immédiatement à droite du mot le mot tampon U et l unités (opérateur $\Phi\,(a^{\downarrow} \to aU\underbrace{1\ldots1}_{l}{}^{\downarrow}))$;

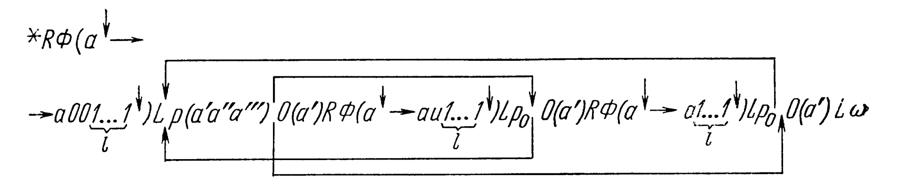
12) On retourne à l'origine du mot (opérateur L) et on passe à 4).

Régime III (le premier symbole a' est le dernier dans le code principal).

13) On efface le symbole a' $(0\,(a'))$;

14) On se déplace à droite jusqu'à l'origine du code l-multiple construit et on s'arrête.

Deuxième étape. On obtient le circuit opératoriel suivant:



La réalisation des opérateurs de ce circuit ne pose aucun problème et l'on peut construire facilement un programme correspondant à ce circuit. Ceci achève la démonstration du lemme.

Lemme 3 (de conversion d'un code réticulé en code principal). *Soit s un entier naturel, $s \geqslant 2$. On peut alors construire une machine de Turing qui convertit un code réticulé quelconque de paramètre s en un code principal.*

Démonstration. *Première étape.* Décrivons d'abord le principe de fonctionnement de la machine. La machine observe à l'instant initial l'unité de gauche du code réticulé, se déplace à gauche d'une certaine distance et se met de proche en proche de droite à gauche à former le code principal par « transfert » des codes des réseaux, en commençant par le s-ième réseau et en finissant par le premier (voir fig. 29).

Cette conversion peut être caractérisée de façon plus précise. A cet effet décomposons-la en une suite d'opérations plus simples

$$*A_0A_s\ldots A_1\omega.$$

Opération A_0 (préparation préalable de la bande). 1) On s'éloigne du champ initial (l'unité de gauche du code réticulé) de s champs à gauche (opérateur L_1^s) *).

2) Immédiatement à gauche de cet intervalle on inscrit une unité (opérateur Φ $(a^{\downarrow} \rightarrow 1^{\downarrow}a)$). L'opération est achevée.

Opération A_s (transfert de la zone d'unités du s-ième réseau). 1) A partir du champ dans lequel l'opérateur A_0 a inscrit l'unité,

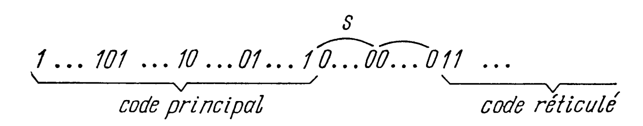


Fig. 29

on se déplace vers la droite de $2s$ champs (opérateur R_1^{2s}). On arrive à l'extrémité gauche de la zone située sur le s-ième réseau. Soient a' et a'' les deux premiers symboles du s-ième réseau (a' est l'extrémité gauche).

2) On vérifie si a' n'est pas en même temps extrémité droite du code par le calcul du prédicat $\tilde{p}$ (a', a'') (en retournant à a'):

$$\tilde{p}(a', a'') = \begin{cases} 0 \text{ pour } a'' = 0 \ (a' \text{ est l'extrémité droite de la zone}), \\ 1 \text{ pour } a'' = 1 \ (a' \text{ n'est pas l'extrémité droite de la zone}). \end{cases}$$

Si $\tilde{p} = 1$ (a' n'est pas l'extrémité droite de la zone du s-ième réseau), alors

3) On se déplace jusqu'à l'extrémité droite de la zone sur ce réseau (opérateur $\tilde{R}$).

4) On efface le dernier symbole a dans la zone (opérateur $0\,(a)$).

5) On retourne à l'extrémité gauche de la zone (opérateur $\tilde{L}$).

6) On se déplace à gauche de $2s$ champs (opérateur L_1^{2s}) et on arrive à l'extrémité droite du code principal (plus exactement de la portion de code construite).

7) On passe à l'extrémité gauche du code principal (opérateur L).

8) On inscrit le symbole 1 à gauche du code principal (opérateur Φ $(a^{\downarrow} \rightarrow 1^{\downarrow}a)$).

9) On retourne à l'extrémité droite du code principal (opérateur R), puis on passe à 1).

Si $\tilde{p} = 0$ (a' est l'extrémité droite de la zone sur le s-ième réseau), alors

*) Le code réticulé peut contenir $s - 1$ zéros de suite.

10) On efface le symbole a' (opérateur $0\,(a')$).

11) En se déplaçant à gauche de $2s$ champs (opérateur L_1^{2s}), on arrive à l'extrémité droite du code principal.

12) On accède à l'extrémité gauche du code principal (opérateur L).

13) On inscrit à gauche du code principal les symboles 10 (opérateur $\Phi\,(a^{\downarrow} \to 1^{\downarrow}0a)$).

14) On retourne à l'extrémité droite du code principal (opérateur R).

L'opération est finie.

Opération A_i $(1 < i \leqslant s)$ (transfert de la zone d'unités du i-ième réseau). Procéder comme pour A_s en remplaçant les opérateurs R_1^{2s} et L_1^{2s} respectivement par R_1^{s+i} et L_1^{s+i}. Ceci nous conduit au i-ième réseau.

Opération A_1 (transfert de la zone d'unités du premier réseau). Procéder comme pour A_s en remplaçant les opérateurs R_1^{2s} et L_1^{2s} respectivement par R_1^{s+1} et L_1^{s+1} et en retirant les opérateurs $\Phi\,(a^{\downarrow} \to 1^{\downarrow}0a)$ et R (voir 13) et 14)), puisque ces opérateurs assurent la préparation de l'opération suivante et A_1 est la dernière.

Deuxième étape. On a de toute évidence les circuits opératoriels suivants pour $A_0, A_s, \ldots, A_i, \ldots, A_1$:

$$A_0 = *L_1^s \Phi(a^{\downarrow} \to 1^{\downarrow}a)\,\omega,$$

$$A_s = *\,{}^{\downarrow}R_1^{2s}\tilde{p}(a'a'')\,\tilde{R}\,0(a)\,\tilde{L}L_1^{2s}L\Phi(a^{\downarrow} \to 1^{\downarrow}a)Rp_0\,{}^{\uparrow}0(a')L_1^{2s}L\Phi(a^{\downarrow} \to 1^{\downarrow}0a)R\omega,$$

. . . .

$$A_i = *\,{}^{\downarrow}R_1^{s+i}\tilde{p}(a'a'')\,\tilde{R}\,0(a)\,\tilde{L}L_1^{s+i}L\Phi(a^{\downarrow} \to 1^{\downarrow}a)Rp_0\,{}^{\uparrow}0(a')L_1^{s+i}L\Phi(a^{\downarrow} \to 1^{\downarrow}0a)R\omega,$$

. . . .

$$A_1 = *\,{}^{\downarrow}R_1^{s+1}\tilde{p}(a'a'')\,\tilde{R}\,0(a)\,\tilde{L}L_1^{s+1}L\Phi(a^{\downarrow} \to 1^{\downarrow}a)Rp_0\,{}^{\uparrow}0(a')L_1^{s+1}L\omega.$$

Ce qui prouve le lemme.

Lemme 4 (de conversion du code quasi principal en code principal). *Pour tout nombre naturel* l $(l \geqslant 2)$ *on peut construire une machine de Turing qui convertit un code quasi principal arbitraire* $b_1U_1b_2 \ldots \ldots U_{\nu-1}b_\nu$, *où* $|U_1| = \ldots = |U_{\nu-1}| = l - 1$, *en le code principal correspondant* $b_1b_2 \ldots b_\nu$.

Démonstration. *Première étape.* La machine cherchée se déplace d'abord d'un certain intervalle à droite du code quasi principal et ensuite forme petit à petit le code principal par transfert du code principal du réseau.

Plus exactement, la conversion se déroule comme suit.

1) A l'instant initial on repère le symbole b_1, c'est-à-dire le symbole situé dans un champ du réseau de pas l contenant le code principal. On se déplace sur le réseau vers l'extrémité droite du code principal (opérateur $\widetilde{R}$) et on efface les mots tampons $U_1, \ldots, U_{\nu-1}$ extérieurs au réseau (voir corollaire 2 du lemme 1).

2) A droite de l'extrémité droite (i.e. du symbole b_ν) on inscrit $3l - 1$ zéros *) et une unité (opérateur $\Phi\,(b_\nu^{\downarrow} \to b_\nu \underbrace{0 \ldots 0}_{3l-1}1^{\downarrow})$).

On se trouve dans un champ de ce réseau.

3) On cherche l'extrémité gauche du code principal (opérateur $\widetilde{L}$).

4) Soient $a'a''a'''$ les trois premiers symboles du code principal (à l'instant initial ce sont $b_1b_2b_3$). On calcule le prédicat $\widetilde{p}\,(a'a''a''')$ trivalent. Pour $a'' = 1$ (régime principal) on passe à la réalisation de l'opérateur consécutif à $\widetilde{p}$. Pour $a'' = 0$ et $a''' = 1$ le symbole a' est la dernière unité de la zone et puisque $a''' = 1$, il existe encore au moins une zone. Dans ce cas on passe à la réalisation de l'opérateur auquel aboutit la flèche marquée par $a'' = 0$, $a''' = 1$. Pour $a'' = a''' = 0$, le symbole a' est la dernière unité du code principal (c'est-à-dire que $a' = b_\nu$) et on passe alors à la réalisation de l'opérateur vers lequel part la flèche marquée par $a'' = a''' = 0$.

— Régime principal $a'' = 1$.

5) On efface le symbole a' (opérateur $0\,(a')$).

6) On se dirige vers l'extrémité droite du code principal (opérateur $\widetilde{R}$).

7) On se déplace vers la droite de $3l$ champs (opérateur R_1^{3l}).

8) On accède à l'extrémité droite du code formé (opérateur R).

9) On inscrit une unité à droite (opérateur $\Phi\,(a^{\downarrow} \to a1^{\downarrow})$).

10) On se déplace vers l'extrémité gauche du code formé (opérateur L).

— Régime $a'' = 0$, $a''' = 1$.

On procède comme dans le régime précédent, sauf 9) qui est remplacé par la réalisation de l'opérateur $\Phi\,(a^{\downarrow} \to a01^{\downarrow})$.

— Régime $a'' = a''' = 0$.

---

*) Le code quasi principal peut contenir $2l - 1$ zéros de suite.

On efface le symbole a' (opérateur $0\,(a')$) et, en se déplaçant vers la droite de $3l$ champs (opérateur R_1^{3l}), on arrive à l'extrémité gauche du code principal cherché et on s'arrête.

Deuxième étape. On a le schéma opératoriel suivant:

$$*\tilde{R}\Phi(b_\nu \to b_\nu \underbrace{0\ldots0}_{3l-1}1)$$

$$a''=0,\ a'''=1$$

$$\tilde{L}\,p(a'a''a''')\ 0(a')\tilde{R}R_1^{3l}\ R\Phi(a \to a1)\,L\,p_0\ 0(a')\tilde{R}R_1^{3l}\ R\Phi(a \to a01)\,L\,p_0\ 0(a')R_1^{3l}\,\omega$$

$$a''=a'''=0$$

§ 22. Fonctions calculables

Nous avons introduit plus haut un système $P_{\aleph_0}$ contenant toutes les constantes de $E^{\aleph_0}$ et toutes les fonctions définies sur l'ensemble des combinaisons de $E^{\aleph_0}$ et à valeurs sur $E^{\aleph_0}$. Nous allons définir un système de fonctions plus large que $P_{\aleph_0}$.

Soit $f(x_1, \ldots, x_n)$ une fonction définie sur un sous-ensemble E_f de l'ensemble des combinaisons $(\alpha_1, \ldots, \alpha_n)$ des nombres de $E^{\aleph_0}$ et à valeurs sur $E^{\aleph_0}$ (on admet que la fonction f n'est pas définie à l'extérieur de l'ensemble E_f). La fonction f s'appelle *fonction partielle de la logique $\aleph_0$-valente.* Désignons par $P_{\aleph_0}^{\mathrm{p}}$ l'ensemble de toutes les fonctions partielles de la logique $\aleph_0$-valente.

On peut introduire la notion de variable inessentielle comme dans le cas d'une fonction non partout définie de P_2 (voir chapitre 1).

Définition. On dit qu'une variable x_i est *inessentielle* pour une fonction $f(x_1, \ldots, x_n)$ de $P_{\aleph_0}^{\mathrm{p}}$ s'il existe une fonction f' de $P_{\aleph_0}$ telle que

$$f'(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$$

sur E_f et la variable x_i est inessentielle pour $f'(x_1, \ldots, x_n)$.

Dans la suite, les fonctions partielles seront considérées aux variables inessentielles près par rapport auxquelles l'ensemble E_f est cylindrique. Dans ce cas, la fonction f se prolonge à une fonction f' de $P_{\aleph_0}$ qui dépend inessentiellement de toutes ces variables.

Définition. On dit qu'une fonction $f(x_1, \ldots, x_n)$, où $f \in P_{\aleph_0}^{\mathrm{p}}$, est *calculable* s'il existe une machine de Turing $\mathfrak{M}$ qui:

a) traitant le code principal d'une combinaison $(\alpha_1, \ldots, \alpha_n) \in E_f$ et observant l'unité gauche de ce code à l'état initial, s'arrête et délivre à l'état final le code de $f(\alpha_1, \ldots, \alpha_n)$;

b) traitant le code principal d'une combinaison $(\alpha_1, \ldots, \alpha_n) \notin \notin E_f$, soit ne s'arrête pas, soit s'arrête, mais l'enregistrement délivré n'est celui d'aucun nombre de $E^{\aleph_0}$.

Remarque. Les constantes de $E^{\aleph_0}$ peuvent également être considérées comme des fonctions calculables dont l'ensemble des variables est vide au sens suivant.

Soit $\gamma \in E^{\aleph_0}$. Considérons la machine définie par le tableau 32. Comme γ est une constante, la bande est supposée vide à l'instant

Tableau 32

	$\varkappa_1$	$\varkappa_2$	...	$\varkappa_{\gamma+1}$	$\varkappa_{\gamma+2}$
0	$1R\varkappa_2$	$1R\varkappa_3$	...	$1S\varkappa_{\gamma+2}$	$0R\varkappa_{\gamma+1}$
1					$1L\varkappa_{\gamma+2}$

Tableau 33

	$\varkappa_1$	$\varkappa_2$
0	$1S\varkappa_2$	
1	$0R\varkappa_1$	

initial. Il est évident que la machine se dirige vers la droite à partir du champ initial et forme une zone de $\gamma + 1$ unités (le code γ), puis revient au début de la zone.

Citons un exemple de fonction calculable.

Exemple 6. Montrons que la fonction $0(x) \equiv 0$ est calculable. Considérons pour cela la machine définie par le tableau 33. Il est évident que cette machine réalise la fonction $0(x) \equiv 0$. On remarquera que cette machine réalise aussi la fonction $f(x_1, x_2) = x_2 + 1$ et la constante 0 (traitée comme une fonction dépendant d'un ensemble vide de variables).

Désignons par P_{cal} la classe de toutes les fonctions calculables. Il est évident que $P_{\text{cal}} \subseteq P^{\text{p}}_{\aleph_0}$.

Définition. Une machine de Turing $\mathfrak{M}$ réalise (calcule) *correctement* une fonction $f(x_1, \ldots, x_n)$ (de la classe P_{cal}) si :

a) traitant le code principal d'une combinaison $(\alpha_1, \ldots, \alpha_n) \in \in E_f$ et se trouvant à l'état initial sur l'unité gauche de ce code, s'arrête et délivre à l'état final le code de $f(\alpha_1, \ldots, \alpha_n)$, cet arrêt ayant lieu sur l'unité gauche du code de $f(\alpha_1, \ldots, \alpha_n)$;

b) traitant le code principal d'une combinaison $(\alpha_1, \ldots, \alpha_n) \notin \notin E_f$ ne s'arrête pas.

Il est aisé de voir que la machine (voir tableau 33) réalise correctement la fonction $0(x) \equiv 0$.

Lemme 5. *Si $f(x_1, \ldots, x_n)$ est une fonction calculable, il existe une machine de Turing qui la calcule correctement.*

Démonstration. Soit $\mathfrak{M}'$ une machine calculant la fonction $f(x_1, \ldots, x_n)$. Désignons par A' un opérateur modifiant l'enregistrement de la bande, la position de la tête et les états de la machine. Considérons la conversion

$$A = * K_1 \overset{\downarrow}{\tilde{A}'} \overset{\urcorner}{p} 0_2 K_3 \omega .$$

Ici K_1 convertit le code de $(\alpha_1, \ldots, \alpha_n)$ en un code double avec 01 pour mot tampon. Sur le premier réseau on aura le code de $(\alpha_1, \ldots$ $\ldots, \alpha_n)$, sur le second, une zone remplie d'unités.

$\tilde{A}'$ est un opérateur simulant A' sur le premier réseau ; en dehors de ce réseau $\tilde{A}'$ inscrit le symbole 1 dans la zone de travail.

p est un prédicat définissant la forme du mot sur le premier réseau après l'action de l'opérateur $\tilde{A}'$. L'inspection du mot est effectuée avec le deuxième réseau qui marque avec sa zone d'unités la zone étudiée sur le premier réseau. On pose $p = 1$ si le mot est une zone d'unités, et $p = 0$ si le mot contient deux unités encadrant un zéro ou s'il ne contient pas d'unités du tout.

0_2 est un opérateur qui efface toutes les unités du deuxième réseau et stoppe la machine sur l'unité de gauche du mot situé dans le premier réseau.

K_3 convertit le code quasi principal en un code principal.

Ce schéma nous apprend que si après l'action de $\tilde{A}'$ l'enregistrement du premier réseau est différent de la zone d'unités, alors l'opération deviendra cyclique, puisque le prédicat p sera constamment calculé.

Tableau 34

s	$\varkappa_1$	$\varkappa_2$
0	$1S\varkappa_2$	
1	$1L\varkappa_1$	

La machine $\mathfrak{M}$ correspondant à la conversion A est la machine cherchée. Le lemme est démontré.

Dans la suite, quand on aura affaire à des fonctions calculables on utilisera exclusivement des machines les calculant correctement.

Passons maintenant à la description de quelques fonctions calculables élémentaires. Considérons les fonctions suivantes : 1) la constante 0 ; 2) $S(x) = x + 1$; 3) $I_m^n(x_1, \ldots, x_n) = x_m$, où $1 \leqslant$ $\leqslant m \leqslant n$.

Tableau 35

I_m^n	$\varkappa_1$	...	$\varkappa_{m-1}$	$\varkappa_m$	$\varkappa_{m+1}$	...	$\varkappa_n$	$\varkappa_{n+1}$	$\varkappa_{n+2}$	$\varkappa_{n+3}$
0	$0R\varkappa_2$	...	$0R\varkappa_m$	$0R\varkappa_{m+1}$	$0R\varkappa_{m+2}$	...	$0L\varkappa_{n+1}$	$0L\varkappa_{n+1}$	$0R\varkappa_{n+3}$	
1	$0R\varkappa_1$	...	$0R\varkappa_{m-1}$	$1R\varkappa_m$	$0R\varkappa_{m+1}$	...	$0R\varkappa_n$	$1L\varkappa_{n+2}$	$1L\varkappa_{n+2}$	

Montrons que ces fonctions sont calculables. Ceci a déjà été fait pour la constante 0. La calculabilité des fonctions $S(x)$ et I_m^n résulte de leur réalisabilité par les machines suivantes (voir tableaux 34 et 35). Pour $I_m^n(x_1, \ldots, x_n)$ l'œil se déplace vers la droite (à l'état initial il observe comme toujours l'unité gauche du code principal), et la machine efface toutes les zones du code principal pour $(\alpha_1, \ldots, \alpha_n)$, sauf la m-ième, puis revient à gauche et stoppe sur l'unité gauche de la zone restante.

§ 23. Opérations S, Rp et μ

Définissons sur l'ensemble $P_{\aleph_0}^{\mathrm{p}}$ les trois opérations suivantes: la superposition S, la récursion primitive Rp et la minimisation μ.

La *superposition* s'introduit comme dans les systèmes fonctionnels précédents: on définit d'abord la notion de formule $\mathfrak{A}(x_1, \ldots, x_n)$ sur un système de fonctions donné de $P_{\aleph_0}^{\mathrm{p}}$, puis à toute formule $\mathfrak{A}$ on associe une fonction $f_{\mathfrak{A}}(x_1, \ldots, x_n)$ de $P_{\aleph_0}^{\mathrm{p}}$. Ceci étant, si l'une des fonctions de $\mathfrak{A}$ n'est pas définie sur une combinaison $(\alpha_1, \ldots, \alpha_n)$, on admet que $f_{\mathfrak{A}}(\alpha_1, \ldots, \alpha_n)$ ne l'est pas non plus. Plus exactement, soit

$$\Phi(x_1, \ldots, x_n) = f(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)).$$

Prenons une combinaison quelconque $(\alpha_1, \ldots, \alpha_n)$ de nombres de $E^{\aleph_0}$. Si les fonctions $f_1, \ldots, f_m$ sont définies sur cette combinaison et si la fonction f l'est sur la combinaison $(f_1(\alpha_1, \ldots, \alpha_n), \ldots, f_m(\alpha_1, \ldots, \alpha_n))$, alors la fonction Φ est définie sur $(\alpha_1, \ldots, \alpha_n)$ et $\Phi(\alpha_1, \ldots, \alpha_n) = f(f_1(\alpha_1, \ldots, \alpha_n), \ldots, f_m(\alpha_1, \ldots, \alpha_n))$; sinon Φ n'est pas définie sur la combinaison $(\alpha_1, \ldots, \alpha_n)$.

L'opération de *récursion primitive* se définit comme suit.

Soient $\varphi(x_1, \ldots, x_n)$ et $\psi(x_1, \ldots, x_n, x_{n+1}, x_{n+2})$*) des fonctions arbitraires de $P_{\aleph_0}^{\mathrm{p}}$. Formons la fonction $f(x_1, \ldots, x_n, x_{n+1})$ en utilisant le « schéma » de la récursion primitive

$$\begin{cases} f(x_1, \ldots, x_n, 0) = \varphi(x_1, \ldots, x_n), \\ f(x_1, \ldots, x_n, y+1) = \psi(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)). \end{cases}$$

Soit $(\alpha_1, \ldots, \alpha_{n+1})$ une combinaison quelconque de nombres de $E^{\aleph_0}$. Posons

$$f(\alpha_1, \ldots, \alpha_n, 0) = \varphi(\alpha_1, \ldots, \alpha_n).$$

Si φ n'est pas définie sur cette combinaison, on admet que $f(\alpha_1, \ldots, \alpha_n, 0)$ ne l'est pas non plus, de même que $f(\alpha_1, \ldots, \alpha_n, y)$ pour

*) Certaines des variables de φ et ψ sont susceptibles d'être absentes.

tout y. Dans le cas contraire on admet que

$$f(\alpha_1, \ldots, \alpha_n, 1) = \psi(\alpha_1, \ldots, \alpha_n, 0, f(\alpha_1, \ldots, \alpha_n, 0)).$$

Si le second membre n'est pas défini, on admet que $f(\alpha_1, \ldots, \alpha_n, 1)$ et $f(\alpha_1, \ldots, \alpha_n, y)$ ne sont pas définies pour tout y, $y \geqslant 1$, etc.

Au bout d'un nombre fini de pas, soit nous déterminons $f(\alpha_1, \ldots, \alpha_n, \alpha_{n+1})$, soit nous établissons que la fonction f n'est pas définie sur cette combinaison.

Ce raisonnement montre que si $f(\alpha_1, \ldots, \alpha_n, \alpha_{n+1})$ n'est pas définie, alors la fonction $f(\alpha_1, \ldots, \alpha_n, \beta)$ ne le sera pas non plus pour $\beta \geqslant \alpha_{n+1}$. S'agissant de la fonction f on dira qu'elle a été obtenue à partir des fonctions φ et ψ par une récursion primitive.

Exemple 7. Montrons que la fonction $f(x_1, x_2) = x_1 + x_2$ peut être obtenue par une récursion primitive à partir de fonctions calculables élémentaires.

En effet

$$\begin{cases} f(x_1, 0) = I_1^1(x_1), \\ f(x_1, y+1) = S(f(x_1, y)). \end{cases}$$

Remarque. L'opération Rp permet d'introduire des variables inessentielles pour toute fonction $\varphi(x_1, \ldots, x_n)$, plus exactement

$$\begin{cases} f(x_1, \ldots, x_n, 0) = \varphi(x_1, \ldots, x_n), \\ f(x_1, \ldots, x_n, y+1) = \varphi(x_1, \ldots, x_n). \end{cases}$$

L'opération de *minimisation* se définit comme suit. Soit $\varphi(x_1, \ldots, x_{n-1}, x_n)$ une fonction quelconque de $P_{\aleph_0}^{\mathfrak{P}}$. Formons la fonction $f(x_1, \ldots, x_{n-1}, x_n)$ au moyen de l'opérateur de minimisation

$$f(x_1, \ldots, x_n) = \mu_y(\varphi(x_1, \ldots, x_{n-1}, y) = x_n),$$

ce qui signifie que pour toute combinaison $(\alpha_1, \ldots, \alpha_n)$ on compose l'équation

$$\varphi(\alpha_1, \ldots, \alpha_{n-1}, y) = \alpha_n.$$

a) S'il existe un y de $E^{\aleph_0}$ qui est solution de cette équation, on prend la plus petite des solutions et on la désigne par μ_y. Si les valeurs

$$\varphi(\alpha_1, \ldots, \alpha_{n-1}, 0), \ldots, \varphi(\alpha_1, \ldots, \alpha_{n-1}, \mu_y - 1)$$

sont également définies, on admet que

$$f(\alpha_1, \ldots, \alpha_{n-1}, \alpha_n) = \mu_y.$$

b) Dans le cas contraire, c'est-à-dire lorsque ou bien l'équation ne possède pas de solutions, ou bien l'une au moins des valeurs $\varphi(\alpha_1, \ldots, \alpha_{n-1}, 0), \ldots, \varphi(\alpha_1, \ldots, \alpha_{n-1}, \mu_y - 1)$ n'est pas définie, la fonction $f(\alpha_1, \ldots, \alpha_n)$ ne l'est pas non plus.

On dit de la fonction f qu'elle se déduit de φ par minimisation.

Exemple 8. Soit $\varphi(x) = x + 1$. Définissons au moyen de l'opération μ la fonction $f(x)$:

$$f(x) = \mu_y (\varphi(y) = x).$$

Il est évident que

$$f(x) = \begin{cases} \text{n'est pas définie} & \text{pour } x = 0, \\ x - 1 & \text{pour } x > 0. \end{cases}$$

Ces opérations permettent de former les trois systèmes fonctionnels suivants.

I. L'ensemble P_{pr} de toutes les fonctions que l'on peut obtenir à partir du système de fonctions $\{0, S(x), I_m^n(x_1, \ldots, x_n), 1 \leqslant m \leqslant n, n = 1, 2, \ldots\}$ par les opérations S, Rp et μ, appelé *classe des fonctions partiellement récursives.*

II. La *classe des fonctions récursives*, c'est-à-dire l'ensemble P_{r} de toutes les fonctions partout définies de P_{pr}.

III. La *classe des fonctions récursives primitives*, i.e. l'ensemble P_{rp} de toutes les fonctions que l'on peut déduire du système $\{0, S(x), I_m^n(x_1, \ldots, x_n), 1 \leqslant m \leqslant n, n = 1, 2, \ldots\}$ par les opérations S et Rp.

Il est évident que

$$P_{\text{rp}} \subseteq P_{\text{r}} \subseteq P_{\text{pr}} \subseteq P_{\aleph_0}^{\text{p}}.$$

L'exemple précédent montre que la classe P_{pr} est essentiellement plus large que la classe P_{r}. On montre que la classe P_{r} est aussi essentiellement plus large que la classe P_{rp}.

Considérons des exemples de fonctions récursives primitives. Considérons les fonctions

$$\text{Sg}(x), \ \overline{\text{Sg}}(x), \ \left[\frac{x}{2}\right], \ 2^x, \ x_1 \dot{-} x_2 \text{ et } x_1 \cdot x_2,$$

où

$$\text{Sg}(x) = \begin{cases} 0 \text{ pour } x = 0, \\ 1 \text{ pour } x \neq 0, \end{cases} \qquad \overline{\text{Sg}}(x) = \begin{cases} 1 \text{ pour } x = 0, \\ 0 \text{ pour } x \neq 0, \end{cases}$$

$$x_1 \dot{-} x_2 = \begin{cases} 0 & \text{pour } x_1 < x_2, \\ x_1 - x_2 & \text{pour } x_1 \geqslant x_2. \end{cases}$$

Les fonctions $\left[\frac{x}{2}\right]$, 2^x et $x_1 \cdot x_2$ ont leur signification habituelle, c'est-à-dire partie entière de $\frac{x}{2}$, exponentielle et multiplication.

Leur récursibilité primitive découle des relations suivantes:

$$\begin{cases} \mathrm{Sg}(0)=0, \\ \mathrm{Sg}(x+1)=1, \end{cases} \qquad \begin{cases} \overline{\mathrm{Sg}}(0)=1, \\ \overline{\mathrm{Sg}}(x+1)=0, \end{cases}$$

$$\begin{cases} x_1\cdot 0=0, \\ x_1(x_2+1)=x_1x_2+x_1, \end{cases} \qquad \begin{cases} 2^0=1, \\ 2^{x+1}=2\cdot 2^x. \end{cases}$$

La constante 1 s'obtient par superposition de 0 et de $S(x)$; x_1+x_2 est récursive primitive, $2x$ s'obtient à partir d'elle par superposition

$$\begin{cases} 0 \dot{-} 1=0, \\ (x+1)\dot{-}1=x, \end{cases} \qquad \begin{cases} x_1\dot{-}0=x_1, \\ x_1\dot{-}(x_2+1)=(x_1\dot{-}x_2)\dot{-}1, \end{cases}$$

$$\begin{cases} \left[\frac{0}{2}\right]=0, \\ \left[\frac{x+1}{2}\right]=x\dot{-}\left[\frac{x}{2}\right], \end{cases}$$

où $x_1\dot{-}1$ est une fonction auxiliaire.

Les fonctions récursives primitives données permettent de construire beaucoup d'autres fonctions récursives primitives. Par exemple, $f(x_1, \ldots, x_n)$, qui est nulle sauf en un nombre fini de points en lesquels ses valeurs appartiennent à $E^{\aleph_0}$, est récursive primitive.

En effet, soit

$$f(x_1, \ldots, x_n)=\begin{cases} \beta^i \text{ pour } (x_1, \ldots, x_n)=(\alpha_1^i, \ldots, \alpha_n^i)\ (i=1, \ldots, s) \\ 0 \text{ dans les autres cas} \end{cases}$$

et $\beta^i\in E^{\aleph_0}\ (i=1, \ldots, s)$.

Considérons les fonctions $j_i(x)$, où

$$j_i(x)=\begin{cases} 1 \text{ pour } x=i, \\ 0 \text{ pour } x\neq i, \end{cases} \qquad i=0, 1, \ldots$$

Il est évident que

$$j_i(x)=\mathrm{Sg}(x\dot{-}(i-1))\cdot\overline{\mathrm{Sg}}(x\dot{-}i) \text{ pour } i\neq 0 \text{ et}$$

$$j_0(x)=\overline{\mathrm{Sg}}(x).$$

Posons

$$j_{i_1, \ldots, i_n}(x_1, \ldots, x_n)=\begin{cases} 1 \text{ pour } (x_1, \ldots, x_n)=(i_1, \ldots, i_n), \\ 0 \text{ dans les autres cas.} \end{cases}$$

On a

$$j_{i_1, \ldots, i_n}(x_1, \ldots, x_n)=j_{i_1}(x_1)\ldots j_{i_n}(x_n),$$

$$f(x_1, \ldots, x_n)=\sum_{i=1}^{s}\beta^i j_{\alpha_1^i\ldots\alpha_n^i}(x_1, \ldots, x_n).$$

La dernière expression est l'analogue du développement en une forme normale disjonctive.

Remarquons en conclusion que les opérations S et Rp appliquées à des fonctions partout définies donnent des fonctions partout définies. Il s'ensuit que la classe P_{r} est fermée pour ces opérations.

Nous allons maintenant étudier le lien existant entre les classes P_{cal} et P_{pr}, et de façon plus précise établir l'identité de ces classes.

§ 24. Fonctions calculables et opérations S, Rp, μ

On se propose d'étudier ici les propriétés des opérations S, Rp et μ sur les fonctions calculables.

Comme pour les systèmes fonctionnels précédents nous considérerons les fonctions $f(x_1, \ldots, x_n)$ aux variables inessentielles près, plus exactement à des variables inessentielles d'un type spécial. Ceci est dû au fait que la fonction calculable $f(x_1, \ldots, x_n)$ est définie sur un ensemble E_f qui n'est pas nécessairement confondu avec l'ensemble de toutes les combinaisons $(\alpha_1, \ldots, \alpha_n)$ de nombres de $E^{\aleph_0}$. Dans ce cas si l'on traite une variable inessentielle, par analogie avec les fonctions de P_k, comme une variable dont la fonction ne dépend pas pour les combinaisons de E_f, on se heurte à certaines difficultés. Par exemple, l'élimination des variables inessentielles peut ne pas se faire d'une manière unique. (Voir raisonnement correspondant pour les fonctions non partout définies de P_k dans le chapitre 3.) Pour cette raison, on dira qu'une variable x_i d'une fonction $f(x_1, \ldots, x_n)$ de P_{cal} est *inessentielle* (au sens strict) si : 1) E_f est cylindrique par rapport à x_i ; 2) f ne dépend pas de x_i sur les combinaisons de E_f.

Explicitons l'élimination d'une variable inessentielle (voir page 11). Supposons par souci de simplicité que x_n est une variable inessentielle de la fonction $f(x_1, \ldots, x_n)$. Par définition, E_f est un ensemble cylindrique par rapport à x_n, et f ne dépend pas de x_n sur les combinaisons de E_f. Considérons une fonction $g(x_1, \ldots, x_{n-1})$ de domaine de définition E_g telle que :

1) E_g soit la projection de E_f sur le sous-espace $(x_1, \ldots, x_{n-1})$; ceci, en vertu de la cylindricité de E_f par rapport à x_n, est équivalent à la condition $(\alpha_1, \ldots, \alpha_{n-1}) \in E_g$ si et seulement si pour tout $\alpha_n \in E^{\aleph_0}$, $(\alpha_1, \ldots, \alpha_{n-1}, \alpha_n) \in E_f$;

2) pour tout $(\alpha_1, \ldots, \alpha_{n-1})$ de E_g

$$g(\alpha_1, \ldots, \alpha_{n-1}) = f(\alpha_1, \ldots, \alpha_{n-1}, 0).$$

On introduit une variable inessentielle de la manière suivante. Soit $f(x_1, \ldots, x_n)$ une fonction calculable de domaine de définition E_f. Considérons une fonction $h(x_1, \ldots, x_n, x_{n+1})$ définie sur E_h et telle que :

1) E_h est un cylindre par rapport à x_{n+1} de base E_f, c'est-à-dire que $(\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) \in E_h$ si et seulement si $(\alpha_1, \ldots, \alpha_n) \in E_f$;

2) pour tout $(\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) \in E_h$

$$h(\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) = f(\alpha_1, \ldots, \alpha_n).$$

Lemme 6. *Si l'on ajoute ou élimine des variables inessentielles d'une fonction calculable on obtient une fonction calculable.*

Démonstration. Prouvons ce lemme sur des cas particuliers.

a) Supposons que $g(x_1, \ldots, x_{n-1})$ se déduit à partir de $f(x_1, \ldots, x_{n-1}, x_n)$ par élimination de la variable inessentielle x_n. Considérons la conversion

$$\text{code de } (\alpha_1, \ldots, \alpha_{n-1}) \to \text{code de } (\alpha_1, \ldots, \alpha_{n-1}, 0) \to \to \text{code de } f(\alpha_1, \ldots, \alpha_{n-1}, 0).$$

La machine de Turing correspondante calcule de toute évidence la fonction $g(x_1, \ldots, x_{n-1})$.

b) Supposons que $h(x_1, \ldots, x_n, x_{n+1})$ a été déduite à partir de $f(x_1, \ldots, x_n)$ par adjonction de la variable inessentielle x_{n+1}. Considérons la conversion

$$\text{code de } (\alpha_1, \ldots, \alpha_n, \alpha_{n+1}) \to \text{code de } (\alpha_1, \ldots, \alpha_n) \to \to \text{code de } f(\alpha_1, \ldots, \alpha_n).$$

La machine de Turing correspondante calcule la fonction $h(x_1, \ldots, x_n, x_{n+1})$.

Le cas général se ramène, au moyen d'un corollaire du lemme suivant, à ceux que nous venons de démontrer.

Lemme 7*). *Si*

$$f(x_1, \ldots, x_m), f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)$$

sont des fonctions calculables, alors la fonction

$$f(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$$

l'est également.

Démonstration. Considérons la conversion

$$*K_1 A_1 A_2 K_2 A_3 A_4 \omega.$$

K_1 convertit le code de $(\alpha_1, \ldots, \alpha_n)$ en le code $(m+1)$-multiple de mot tampon $U = \underbrace{0 \ldots 01}_{m+1}$. Il est évident qu'on obtient des codes

*) Dans la démonstration du lemme, il est essentiel que la fonction calculable soit réalisable par une machine qui la calcule correctement.

coïncidant avec le code de $(\alpha_1, \ldots, \alpha_n)$ dans les m premiers réseaux de pas $m+1$, et une zone d'unités dans le $(m+1)$-ième réseau.

A_1 convertit le code de $(\alpha_1, \ldots, \alpha_n)$ en

le code de $f_1(\alpha_1, \ldots, \alpha_n)$ sur le premier réseau,

le code de $f_2(\alpha_1, \ldots, \alpha_n)$ sur le second réseau,

. . . .

le code de $f_m(\alpha_1, \ldots, \alpha_n)$ sur le m-ième réseau

et affiche 1 partout dans le $(m+1)$-ième réseau. On effectue cette conversion en utilisant m fois les machines simulant le calcul des fonctions $f_i(x_1, \ldots, x_n)$ $(i = 1, \ldots, m)$ (voir corollaire 4 du lemme 1).

A_2 procède à un « tassement » des codes des $f_i(\alpha_1, \ldots, \alpha_n)$ sur les réseaux i $(i = 1, \ldots, m)$. Pour cela on trouve dans le $(m+1)$-ième réseau l'unité gauche et on se déplace à gauche d'elle de $3m+2$ champs *). On accède ainsi au premier réseau et on déplace le code de $f_1(\alpha_1, \ldots, \alpha_n)$ vers ce champ (on simule une machine effectuant une translation à gauche). Ensuite, du champ qui contient l'unité gauche dans le premier réseau, on se déplace d'un champ à droite et on se trouve dans le deuxième réseau. De façon analogue, on déplace le code de $f_2(\alpha_1, \ldots, \alpha_n)$ vers ce champ, etc. Après le transfert du code de $f_m(\alpha_1, \ldots, \alpha_n)$ dans le m-ième réseau, l'œil observe l'unité gauche du m-ième réseau. On efface la région du $(m+1)$-ième réseau située à gauche de ce champ et on retourne à l'unité gauche sur le premier réseau. Ceci nous donne un code réticulé de paramètre $s = m+1$.

K_2 convertit le code réticulé en code principal.

A_3 efface la $(m+1)$-ième zone du code principal et fait revenir l'œil de la machine à l'unité gauche. La bande affiche ainsi le code de $(f_1(\alpha_1, \ldots, \alpha_n), \ldots, f_m(\alpha_1, \ldots, \alpha_n))$.

A_4 convertit ce code en le code de $f(f_1(\alpha_1, \ldots, \alpha_n), \ldots \ldots, f_m(\alpha_1, \ldots, \alpha_n))$.

Il est évident que cette conversion est réalisable si et seulement si la valeur de $f(f_1, \ldots, f_m)$ est définie. De fait, on utilise ici la calculabilité correcte des fonctions $f, f_1, \ldots, f_m$ par les machines.

Donc, la machine qui réalise cette conversion est la machine cherchée. Ce qui prouve le lemme.

Corollaire 1. *Soit* $\begin{pmatrix} 1 & 2 & \ldots & m \\ i_1 & i_2 & \ldots & i_m \end{pmatrix}$ *une permutation quelconque. Alors*

$$f(I^m_{i_1}(x_1, \ldots, x_m), I^m_{i_2}(x_1, \ldots, x_m), \ldots, I^m_{i_m}(x_1, \ldots, x_m)) = \\ = f(x_{i_1}, \ldots, x_{i_m}).$$

*) L'unité gauche du $(m+1)$-ième réseau peut se trouver à m champs à droite de l'unité gauche du premier réseau.

Ceci signifie que la fonction déduite à partir d'une fonction calculable par permutation des variables est calculable.

Corollaire 2. *Le lemme 7 se généralise aisément au cas où les fonctions* $f_1, \ldots, f_m$ *ne dépendent pas de toutes les variables* $x_1, \ldots, x_n$.

On obtient ceci par adjonction de toutes les variables inessentielles manquantes (lemme 6) aux fonctions $f_1, \ldots, f_m$ et par application du lemme 7.

Théorème 1. *La classe* P_{cal} *est fermée pour la superposition.*

Démonstration. Elle repose sur le lemme 7 et sur le fait que la fonction identique $I_1^1(x_1)$ appartient à la classe P_{cal} (voir remarque 2 page 16).

L'étude des opérations Rp et μ passe par l'analyse et la conversion de codes situés dans des réseaux de pas l. Considérons de ce fait trois opérateurs que l'on peut caractériser comme suit:

l'opérateur $p_>(\beta, \beta')$ compare deux nombres β et β' $(\beta \geqslant \beta')$ situés dans deux réseaux:

$$p_>(\beta, \beta') = \begin{cases} 1 \text{ pour } \beta > \beta', \\ 0 \text{ pour } \beta = \beta'; \end{cases}$$

l'opérateur $T(i, j)$ transfère (sans l'effacer) le code principal du i-ième réseau dans un réseau « vide » d'indice j;

l'opérateur $T_f(i, j)$ déplace (en l'effaçant) le code donné d'un nombre f du réseau i dans le réseau j et le dispose à un champ vide à gauche du code principal.

Dans la suite, on étudiera des réseaux de pas 3 et 4. Plus bas on démontre des lemmes pour des réseaux de pas 4 et pour des valeurs spéciales des paramètres i et j. Pour les autres cas ces propositions se démontrent de façon analogue.

Lemme 8. *Supposons que l'opérateur* $p_>(\beta, \beta')$ *compare des nombres* β *et* β' $(\beta \geqslant \beta')$ *contenus respectivement dans le premier et le second réseau, l'origine du code de* β' *se trouvant dans le champ contigu à droite de celui qui contient l'origine du code de* β. *Supposons qu'à l'instant initial et à l'instant final l'œil observe le début du code de* β. *Supposons enfin que la machine ne change pas tout l'enregistrement de la bande et s'arrête dans l'état* $\varkappa'$ *si* $p_> = 1$ *et dans l'état* $\varkappa''$ *si* $p_> = 0$. *Il existe alors une machine de Turing réalisant l'opérateur* $p_>(\beta, \beta')$.

Démonstration. Considérons le tableau 36. Il est évident que cette machine s'arrête pour $\beta > \beta'$ dans l'état $\varkappa_2$ et pour $\beta = \beta'$ dans l'état $\varkappa_1$. Pour construire la machine cherchée il est nécessaire de décaler l'œil de la machine donnée à l'état initial. Ceci démontre le lemme.

Tableau 36

	$\varkappa_1$	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$
0			$0R\varkappa_4$	$0R\varkappa_1$
1	$1R\varkappa_2$	$1R\varkappa_3$	$1R\varkappa_4$	$1R\varkappa_1$

Lemme 9. *Supposons que* T (2, 3) *transfère (sans l'effacer) le code principal du deuxième réseau dans le réseau « vide » de numéro* 3. *Ceci étant l'œil observe l'unité gauche du code principal dans le deuxième réseau à l'instant initial, un champ du deuxième réseau à l'instant final et l'enregistrement n'est pas modifié dans les autres réseaux. On peut alors construire une machine de Turing réalisant l'opérateur* T (2, 3).

Démonstration. Considérons le tableau 37. Il définit de toute évidence la machine qui réalise T (2, 3). Après le transfert la machine s'arrête sur le deuxième réseau à droite du code principal dans l'état $\varkappa_9$. Ce qui prouve le lemme.

Tableau 37

	$\varkappa_1$	$\varkappa_2$	$\varkappa_3$	$\varkappa_4$	$\varkappa_5$	$\varkappa_6$	$\varkappa_7$	$\varkappa_8$	$\varkappa_9$
0		$1R\varkappa_3$	$0R\varkappa_4$	$0R\varkappa_5$	$0R\varkappa_6$	$0R\varkappa_7$	$0R\varkappa_8$	$0R\varkappa_9$	
1	$1R\varkappa_2$		$1R\varkappa_4$	$1R\varkappa_5$	$1S\varkappa_1$		$1R\varkappa_8$	$1R\varkappa_9$	$1S\varkappa_1$

Lemme 10. *Supposons que l'opérateur* T_f (3, 1) *transfère le code de* f *(une zone d'unités) du troisième réseau dans le premier et le dispose à un champ vide à gauche du code principal. Ceci étant, à l'instant finitial, l'œil observe l'unité gauche du code principal inscrit dans le premier réseau, à l'instant final l'unité gauche du code, construit dans le premier réseau, efface le troisième réseau et ne modifie pas l'enregistrement du code principal dans le deuxième réseau. Supposons par ailleurs que le quatrième réseau contienne à l'instant initial une zone d'unités*

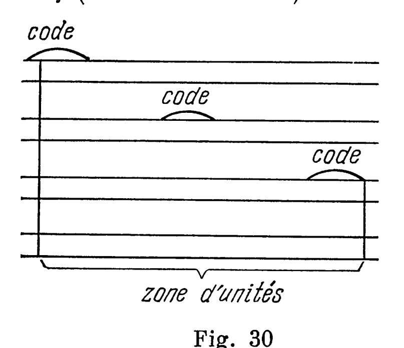


Fig. 30

qui englobe tous les champs de ce réseau dans les limites des codes des trois premiers réseaux (fig. 30), *et qu'à l'instant final le quatrième réseau affiche une zone d'unités englobant aussi tous les champs du*

réseau mais dans les limites des codes construits sur les trois premiers réseaux. Alors on peut construire une machine de Turing réalisant l'opérateur T_f (3, 1).

Démonstration. 1) L'opérateur $\widetilde{\Phi}$ ($a^{\downarrow} \rightarrow 1^{\downarrow}0a$) place les symboles 10 dans le premier réseau à gauche du code principal.

2) On se déplace d'un champ à gauche (opérateur L_1) et on accède au quatrième réseau.

3) L'opérateur $\widetilde{L}_{\mathrm{IV}}$ trouve l'extrémité gauche de la zone d'unités dans le quatrième réseau.

4) On se déplace d'un champ à gauche (opérateur L_1) et on accède au troisième réseau.

5) L'opérateur $\widetilde{L}_{\mathrm{III}}$ trouve l'extrémité gauche du code de f dans le troisième réseau (déplacement à droite).

6) On analyse le début $a'a''$ du code dans le troisième réseau en calculant le prédicat

$$\widetilde{p}(a', a'') = \begin{cases} 0 \text{ pour } a'' = 0 \\ 1 \text{ pour } a'' = 1. \end{cases}$$

Si $\widetilde{p} = 1$, alors:

7) On efface le symbole a' avec l'opérateur 0_{III} (a').

8) On se déplace d'un champ à droite (opérateur R_1), on accède au quatrième réseau.

9) L'opérateur $\widetilde{R}_{\mathrm{IV}}$ trouve l'extrémité droite de la zone d'unités dans le quatrième réseau.

10) L'opérateur R_1 réalise une translation d'un champ à droite.

11) En se déplaçant à gauche dans le premier réseau, on y trouve (opérateur $\widetilde{L}_{\mathrm{I}}$) l'extrémité gauche du code.

12) L'opérateur $\widetilde{\Phi}$ ($a^{\downarrow} \rightarrow 1^{\downarrow}a$) inscrit encore une unité à gauche de ce code dans le premier réseau. Ensuite on revient à l'opérateur 2).

Si $\widetilde{p} = 0$, alors

13) On efface le symbole a' avec l'opérateur 0_{III} (a').

14) On se déplace à droite (opérateur R_2) de deux champs et on aboutit au premier réseau.

15) On cherche (opérateur $\widetilde{L}_1'$) l'extrémité gauche du code principal dans le premier réseau et on s'arrête.

On admet que tous les opérateurs de simulation affichent des unités dans les limites de la zone de travail du quatrième réseau.

Le circuit opératoriel pour T_f(3, 1) est de la forme

$$* \widetilde{\Phi}(a^{\downarrow} \rightarrow 1^{\downarrow} 0a) \uparrow L_1 \widetilde{L}_{\mathrm{IV}} L_1 \widetilde{L}_{\mathrm{III}} p(a'a'') \ulcorner 0_{\mathrm{III}}(a') R_1 \widetilde{R}_{\mathrm{IV}} R_1 \widetilde{L}_{\mathrm{I}} \widetilde{\Phi}(a \rightarrow 1^{\downarrow} a) p_0 \downarrow 0_{\mathrm{III}}(a') R_2 \widetilde{L}_{\mathrm{I}}' \omega$$

Ce qui prouve le lemme.

Théorème 2. *La classe* P_{cal} *est fermée pour l'opération* Rp.

Démonstration. Supposons qu'une fonction $f(x_1, \ldots, x_n, x_{n+1})$ est définie à partir des fonctions calculables $\varphi(x_1, \ldots, x_n)$ et $\psi(x_1, \ldots, x_n, x_{n+1}, x_{n+2})$ à l'aide de l'opération Rp au moyen du schéma

$$\begin{cases} f(x_1, \ldots, x_n, 0) = \varphi(x_1, \ldots, x_n), \\ f(x_1, \ldots, x_n, y+1) = \psi(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)). \end{cases}$$

Montrons que $f(x_1, \ldots, x_n, x_{n+1}) \in P_{\text{cal}}$. Au lieu de ce schéma considérons le schéma suivant

$$\begin{cases} f(x_1, \ldots, x_n, 0) = \varphi(x_1, \ldots, x_n), \\ f(x_1, \ldots, x_n, y+1) = \psi'(f(x_1, \ldots, x_n, y), x_1, \ldots, x_n, y), \end{cases}$$

où

$$\psi'(x_{n+2}, x_1, \ldots, x_n, x_{n+1}) = \psi(x_1, \ldots, x_n, x_{n+1}, x_{n+2}).$$

D'après le corollaire 1 du lemme 7 on a $\psi' \in P_{\text{cal}}$.

1) L'opérateur K_1 convertit le code de $(\alpha_1, \ldots, \alpha_n, \beta)$ en un code quadruple de mot tampon $U = 0001$. Le code obtenu se décompose à l'aide des quatre réseaux de pas 4 en trois exemplaires du code

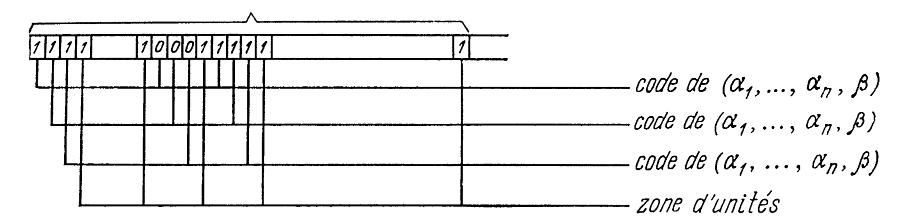


Fig. 31

de $(\alpha_1, \ldots, \alpha_n, \beta)$ situés dans les trois premiers réseaux et en une zone d'unités dans le quatrième réseau (voir fig. 31) située dans les limites des codes des trois premiers réseaux.

2) L'opérateur Φ remplace dans le deuxième réseau le code de β par celui de 0, efface le code de β dans le troisième réseau et stoppe sur l'unité gauche du troisième réseau. Le deuxième réseau affiche alors le code de $(\alpha_1, \ldots, \alpha_n, \beta')$ avec $\beta' = 0$, et le troisième réseau, le code de $(\alpha_1, \ldots, \alpha_n)$.

3) L'opérateur $\tilde{A}_\varphi$ calcule $\varphi(\alpha_1, \ldots, \alpha_n)$ dans le troisième réseau.

4) L'opérateur A_1 cherche l'origine du code de β dans le premier réseau.

5) L'opérateur $p_>$ (β, β') $(\beta \geqslant \beta')$ compare les codes des nombres β et β' situés dans le premier et le deuxième réseau et à l'instant final l'œil observe l'origine du code de β.

Si $p_> = 1$, alors

6) L'opérateur L_{I} cherche l'unité gauche du code dans le premier réseau.

7) L'opérateur T_f (3, 1) transfère le code de $f(\alpha_1, \ldots, \alpha_n, 0)$, qui est égal à celui de $\varphi(\alpha_1, \ldots, \alpha_n)$, du 3-ième réseau dans le premier, le dispose à un champ vide à gauche du code principal, efface le code de f dans le 3-ième réseau et observe l'unité gauche du code principal construit dans le premier réseau, c'est-à-dire le code de $(f, \alpha_1, \ldots, \alpha_n, \beta)$.

8) L'opérateur L_{II} cherche l'unité gauche dans le 2-ième réseau.

9) L'opérateur T (2, 3) transfère le code principal du 2-ième réseau dans le 3-ième (qui est vide) et arrête la machine dans un champ du 2-ième réseau.

10) L'opérateur L_{III} cherche l'unité gauche du code dans le 3-ième réseau.

11) L'opérateur T_f(1, 3) transfère le code de f du premier dans le 3-ième réseau, le place à un champ vide à gauche du code principal (variante du lemme) et affiche le code de $(f, \alpha_1, \ldots, \alpha_n, \beta')$ dans le 3-ième réseau. A l'instant final l'œil observe l'unité gauche de ce code.

12) L'opérateur $\tilde{A}_{\psi'}$ calcule le code de $\psi'(f, \alpha_1, \ldots, \alpha_n, \beta')$ dans le 3-ième réseau. On obtient

$$f(\alpha_1, \ldots, \alpha_n, \beta' + 1) = \psi'(f, \alpha_1, \ldots, \alpha_n, \beta').$$

13) L'opérateur F_+ $(\beta', 1)$ ajoute une unité au code de β' dans le 2-ième réseau et s'arrête sur un champ du 3-ième réseau. On passe ensuite à l'opérateur 4).

Si $p_> = 0$, alors

14) L'opérateur 0 (1, 2, 4) efface les réseaux 1, 2, 4 dans les limites de la zone d'unités du 4-ième réseau et stoppe sur l'unité gauche du 3-ième réseau.

15) L'opérateur K_3 convertit le code quasi principal dans lequel tous les mots tampons sont vides en le code principal. On obtient le code de $f(\alpha_1, \ldots, \alpha_n, \beta)$.

Les opérateurs 1) à 13) sont choisis de manière à ne pas modifier l'enregistrement des autres réseaux et à ajouter des unités dans la zone de travail du 4-ième réseau.

Le circuit opératoriel est de la forme:

$$*K_1 \Phi \tilde{A}_\varphi \, {\uparrow} A_1 p_> (\beta, \beta') \, {}^{\ulcorner} L_{\text{I}} T_f (3,1) L_{\text{II}} T(2,3) L_{\text{III}} T_f (1,3) \tilde{A}_{\psi'} 0_f (1) F_+ (\beta', 1) p_0 \, {}^{\downarrow} 0(1,2,4) K_3 \, \omega$$

Ce qui prouve le théorème.

Remarque. Dans la démonstration du théorème on se sert en fait de la propriété suivante du domaine de définition de la fonction $f(x_1, \ldots, x_{n+1})$: si $f(\alpha_1, \ldots, \alpha_n, \alpha_{n+1})$ n'est pas définie, alors $f(\alpha_1, \ldots, \alpha_n, \beta)$ ne l'est pas pour $\beta \geqslant \alpha_{n+1}$.

Théorème 3. *La classe P_{cal} est fermée pour l'opération μ.*

Démonstration. Soit $\varphi(x_1, \ldots, x_{n-1}, x_n)$ une fonction calculable. Montrons que la fonction $f(x_1, \ldots, x_n)$, où

$$f(x_1, \ldots, x_n) = \mu_y(\varphi(x_1, \ldots, x_{n-1}, y) = x_n)$$

est calculable.

Considérons la fonction auxiliaire

$$\varphi'(y, x_1, \ldots, x_{n-1}) = \varphi(x_1, \ldots, x_{n-1}, y).$$

Le corollaire 1 du lemme 7 nous dit que la fonction φ' est calculable.

1) L'opérateur K_1 convertit le code de $(\alpha_1, \ldots, \alpha_n)$ en un code triple de mot tampon $U = 001$. On obtient un code qui est divisé par trois réseaux de pas 3 en deux exemplaires du code de $(\alpha_1, \ldots, \alpha_n)$ situés dans les deux premiers réseaux et une zone d'unités dans le 3-ième réseau située dans les limites des codes des deux premiers réseaux.

2) L'opérateur $\widetilde{\Phi}_{\text{I,II}}\,(a{\downarrow} \to 1{\downarrow}0a)$ inscrit le code de 0 à gauche des codes principaux dans le premier et le deuxième réseau.

3) L'opérateur $0_{\text{II}}\,(\alpha_n)$ efface la dernière zone, c'est-à-dire le code de α_n, dans le deuxième réseau.

4) L'opérateur $\widetilde{A}_{\varphi'}$ calcule $\varphi'(0, \alpha_1, \ldots, \alpha_{n-1})$ dans le deuxième réseau.

5) L'opérateur A_1 « tasse » les codes dans le premier et le deuxième réseaux, c'est-à-dire les dispose de telle sorte que l'extrémité

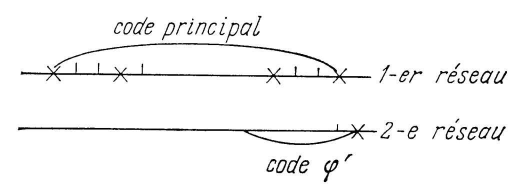


Fig. 32

droite du code de φ' se trouve dans le champ qui suit immédiatement celui qui contient l'extrémité droite du code principal dans le premier réseau (voir fig. 32).

Le tassement des codes peut être réalisé de la manière suivante : en parcourant le troisième réseau on arrive à l'extrémité droite de la

zone d'unités puis on se déplace de 9 champs vers la droite et d'un champ vers la gauche. On arrive ainsi dans un champ du deuxième réseau, champ dans lequel on transfère la zone d'unités (code de φ') (voir exemple 5). Ensuite on se rend dans le champ contigu à gauche de l'extrémité droite du code de φ', on se retrouve dans un champ du premier réseau, champ dans lequel on transfère le code principal (généralisation de l'exemple 5).

6) L'opérateur $p_{\neq}(\varphi', \alpha_n)$ compare les codes de φ' et de α_n qui sont contenus dans le premier et le deuxième réseaux (on utilise un programme analogue à celui du lemme 8):

Si $p = 1$, c'est-à-dire $\varphi \neq \alpha_n$, alors:

7) L'opérateur $\underset{\sim}{0}$ (2) efface le deuxième réseau.

8) L'opérateur $\widetilde{F}_+ (y, 1)$ ajoute une unité au code de y dans le premier réseau.

9) L'opérateur T (1, 2) transfère le code du premier réseau dans le second et revient à l'opérateur 3).

Si $p = 0$, c'est-à-dire $\varphi = \alpha_n$, alors

10) L'opérateur 0 (2, 3) efface les réseaux 2 et 3.

11) L'opérateur 0 (1) efface toutes les zones d'unités du premier réseau sauf la première.

12) L'opérateur K_3 convertit le code quasi principal en code principal et stoppe la machine.

Le circuit opératoriel est de la forme

$$*K_1 \widetilde{\Phi}_{I,II}(a \to 1\, 0a)\, 0_{II}(\alpha_n)\, \widetilde{A}_{\varphi'}\, A_1 p_{\neq}(\varphi' \alpha_n)\, 0(2) \widetilde{F}_+ (y, 1) T(1,2) p_0\, 0(2,3) 0(1) K_3 \omega$$

Lorsqu'on réalise des opérateurs il est nécessaire de tenir compte de leur compatibilité et de ce qu'ils doivent afficher des unités dans la zone de travail du 3-ième réseau. Ceci prouve le théorème.

Remarque. Dans la démonstration du théorème on utilise en principe le fait que si l'une des valeurs $\varphi(\alpha_1, \ldots, \alpha_{n-1}, 0), \ldots$ $\ldots, \varphi(\alpha_1, \ldots, \alpha_{n-1}, \mu_y - 1)$ n'est pas définie, alors $f(\alpha_1, \ldots, \alpha_n)$ ne l'est pas non plus.

Théorème 4. *La classe P_{cal} est fermée pour le système d'opérations* $R = \{S, \text{Rp}, \mu\}$.

Corollaire. $P_{\text{pr}} \subseteq P_{\text{cal}}$.

Ce théorème permet d'établir qu'une fonction est calculable sans passer par la construction d'une machine de Turing mais en prouvant qu'elle est partiellement récursive.

Exemples: a) De nombreuses fonctions récursives primitives ont été construites à la page 130. D'après le théorème prouvé, elles sont aussi calculables.

b) Soit $f(x) = x^2$. Il est évident $f \in P_{\text{rp}}$, puisque

$$f(0) = 0,\ f(x+1) = f(x) + 2x + 1.$$

Donc, $f \in P_{\text{cal}}$.

§ 25. Formule de Kleene. Récursivité partielle des fonctions calculables. Exemples de systèmes complets

Nous allons commencer ce paragraphe par l'établissement de la récursivité primitive de certaines fonctions.

1. Soit $\rho(x)$ le nombre de digits d'un nombre binaire x. Il est évident que

$$\begin{cases} \rho(0) = 1, \\ \rho(x+1) = \rho(x) + \overline{\text{S}}\text{g}\,(2^{\rho(x)} \dot{-} (x+1)). \end{cases}$$

Donc, $\rho(x) \in P_{\text{rp}}$.

2. Soit $\vartheta_n(x_1, \ldots, x_n)$ un entier naturel dont la représentation binaire est:

$$\underbrace{1\ldots1}_{x_1+1}\ 0\ \underbrace{1\ldots1}_{x_2+1}\ \ldots\ 0\ \underbrace{1\ldots1}_{x_n+1}.$$

La récursivité primitive se prouve par récurrence sur n:

$$\begin{cases} \vartheta_1(0) = 1, \\ \vartheta_1(x_1+1) = 2\vartheta_1(x_1) + 1, \end{cases}$$

$$\begin{cases} \vartheta_n(x_1, \ldots, x_{n-1}, 0) = 4\vartheta_{n-1}(x_1, \ldots, x_{n-1}) + 1, \\ \vartheta_n(x_1, \ldots, x_{n-1}, x_n+1) = 2\vartheta_n(x_1, \ldots, x_{n-1}, x_n) + 1. \end{cases}$$

3. Considérons la fonction $\pi(x_1, x_2)$ (voir tableau 38). Cette fonction s'appelle *fonction de Peano* et sert à numéroter tous les couples (α_1, α_2) de nombres de $E^{\aleph_0}$. Il est évident que

$$\pi(x_1, x_2) = \frac{(x_1+x_2)(x_1+x_2+1)}{2} + x_1 = \left[\frac{(x_1+x_2)(x_1+x_2+1)}{2}\right] + x_1,$$

c'est-à-dire qu'elle est récursive primitive.

4. Les fonctions $\lambda(x)$ et $\mu(x)$ sont reliées à la fonction de Peano $\pi(x_1, x_2)$. Supposons que $\lambda(x) = \pi(0, x) = \left[\frac{x(x+1)}{2}\right]$, c'est-à-dire que la fonction $\lambda(x)$ est définie par la première ligne du tableau 38 pour π. Donc $\lambda(x) \in P_{\text{rp}}$.

Définissons la fonction $\mu(x)$ à l'aide du tableau 39. On voit sur ce tableau comment sont reliées π, λ et μ : les valeurs de la fonction π

Tableau 38

x_1 \ x_2	0	1	2	3	...
0	0	1	3	6	...
1	2	4	7	...	
2	5	8	...		
3	9	...			
...	...				

parcourent $E^{\aleph_0}$ et sont divisées par les diagonales $(x_1 + x_2 = c)$ en portions finies ; si α $(\alpha \in E^{\aleph_0})$ est une valeur arbitraire du tableau pour π, alors $\mu(\alpha)$ représente le numéro c de la diagonale sur laquelle

Tableau 39

x	0	1	2	3	4	5	6	7	8	9	...
$\mu(x)$	0	1	1	2	2	2	3	3	3	3	...

se trouve α et $\lambda(\mu(\alpha))$ est le plus petit des nombres de $E^{\aleph_0}$ situé sur cette diagonale.

D'après ce qui précède on a

$$\begin{cases} \mu(0) = 0, \\ \mu(x+1) = \mu(x) + \overline{\mathrm{Sg}}\{\mu(x) \dot{-} (x \dot{-} \lambda(\mu(x)))\}. \end{cases}$$

Donc, $\mu(x) \in P_{\mathrm{rp}}$.

5. Soit (α_1, α_2) un couple quelconque. Alors $\gamma = \pi(\alpha_1, \alpha_2)$ est son numéro. Désignons par $l(x)$ et $r(x)$ les fonctions qui définissent les composantes α_1 et α_2 du couple (α_1, α_2) d'après son numéro γ. Ainsi, $l(\gamma) = \alpha_1$ et $r(\gamma) = \alpha_2$. Ces fonctions vérifient les identités

$$\pi(l(x), r(x)) \equiv x, \quad l(\pi(x_1, x_2)) \equiv x_1, \quad r(\pi(x_1, x_2)) \equiv x_2.$$

Comme

$$l(x) = x \dot{-} \lambda(\mu(x)), \quad r(x) = \mu(x) \dot{-} l(x),$$

alors $l(x)$, $r(x) \in P_{\text{rp}}$.

6. Les fonctions π, l et r peuvent être généralisées au cas de plusieurs variables. La fonction de Peano $\pi_s(x_1, \ldots, x_s)$ définit le numéro d'un s-uple $(\alpha_1, \ldots, \alpha_s)$ de nombres de $E^{\aleph_0}$, quant aux fonctions $t_1(x), \ldots, t_s(x)$, elles définissent les valeurs de ses composantes, c'est-à-dire les nombres $\alpha_1, \ldots, \alpha_s$, d'après le numéro du s-uple $(\alpha_1, \ldots, \alpha_s)$.

Pour $s = 3$, ces fonctions sont définies comme suit:

$$\pi_3(x_1, x_2, x_3) = \pi(x_1, \pi(x_2, x_3)),$$
$$t_1(x) = l(x), \quad t_2(x) = l(r(x)), \quad t_3(x) = r(r(x)).$$

Il est évident que

$$\pi_3(t_1(x), t_2(x), t_3(x)) = x, \quad t_1(\pi_3(x_1, x_2, x_3)) \equiv x_1,$$
$$t_2(\pi_3(x_1, x_2, x_3)) \equiv x_2, \quad t_3(\pi_3(x_1, x_2, x_3)) \equiv x_3.$$

De la définition il s'ensuit que $\pi_3(x_1, x_2, x_3)$, $t_1(x)$, $t_2(x)$, $t_3(x)$ appartiennent à P_{rp}.

Pour la suite de l'exposé nous aurons besoin d'une méthode de construction des fonctions.

Supposons que $x = (x_1, \ldots, x_n)$ et que les fonctions $f_1(x, y), \ldots$ $\ldots, f_s(x, y)$ sont définies à l'aide du schéma de récursion primitive simultanée:

$$\begin{cases} f_1(x, 0) = \varphi_1(x) \\ \cdots\cdots \\ f_s(x, 0) = \varphi_s(x) \\ f_1(x, y+1) = \psi_1(x, y, f_1(x, y), \ldots, f_s(x, y)), \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ f_s(x, y+1) = \psi_s(x, y, f_1(x, y), \ldots, f_s(x, y)), \end{cases}$$

qui est une généralisation naturelle du schéma de récursion primitive. Utilisons ce schéma pour le cas où $\varphi_1, \ldots, \varphi_s, \psi_1, \ldots, \psi_s$ sont récursives primitives. Montrons que ces fonctions peuvent être déduites à partir des fonctions $\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_s$, ainsi que des fonctions récursives primitives $\pi_s, t_1, \ldots, t_s$ par une superposition et une récursion primitive.

Considérons la fonction

$$f(x, y) = \pi_s(f_1(x, y), \ldots, f_s(x, y));$$

on a

$$f(x, 0) = \pi_s(f_1(x, 0), \ldots, f_s(x, 0)) = \pi_s(\varphi_1(x), \ldots, \varphi_s(x)),$$
$$f(x, y+1) = \pi_s(f_1(x, y+1), \ldots, f_s(x, y+1)) =$$
$$= \pi_s(\psi_1(x, y, f_1(x, y), \ldots, f_s(x, y)), \ldots, \psi_s(x, y, f_1(x, y), \ldots$$
$$\ldots, f_s(x, y))) = \pi_s(\psi_1(x, y, t_1(f(x, y)), \ldots, t_s(f(x, y))), \ldots$$
$$\ldots, \psi_s(x, y, t_1(f(x, y)), \ldots, t_s(f(x, y)))) = \psi(x, y, f(x, y)),$$

où

$$\psi(x, y, z) = \pi_s((\psi_1(x, y, t_1(z), \ldots, t_s(z)), \ldots$$
$$\ldots, \psi_s(x, y, t_1(z), \ldots, t_s(z)))).$$

Donc, la fonction $f(x, y)$ peut être obtenue par une récursion primitive à partir de $\pi_s(\varphi_1, \ldots, \varphi_s)$ et de ψ, c'est-à-dire que $f(x, y) \in P_{\text{rp}}$. Par ailleurs,

$$f_1(x, y) = t_1(f(x, y)), \ldots, f_s(x, y)) = t_s(f(x, y)),$$

donc, $f_1(x, y), \ldots, f_s(x, y) \in P_{\text{rp}}$.

Prouvons maintenant le théorème de représentation d'une fonction calculable (donc de toute fonction partiellement récursive) par des fonctions récursives primitives sous une forme spéciale (analogue du théorème de Kleene).

Théorème 5. *Pour toute fonction calculable $f(x_1, \ldots, x_n)$ il existe des fonctions récursives primitives $F_f(x_1, \ldots, x_n, y)$ et $G_f(x_1, \ldots, x_n, y)$, telles que*

$$f(x_1, \ldots, x_n) = F_f(x_1, \ldots, x_n, \mu_y(G_f(x_1, \ldots, x_n, y) = 0)).$$

Démonstration. Soit $f(x_1, \ldots, x_n)$ une fonction calculable quelconque que nous noterons brièvement $f(x)$, où $x = (x_1, \ldots, x_n)$. Considérons la machine de Turing qui calcule cette fonction correctement. Soient T un programme de la machine et $\varkappa_1, \ldots, \varkappa_r$ des états de la machine. Introduisons un nouvel état $\varkappa_0$ (le symbole $\varkappa_0$ est différent de $\varkappa_1, \ldots, \varkappa_r$) et complétons le programme T en remplissant toutes ses cases vides, ainsi que les cases de la colonne associée $\varkappa_0$ de la manière suivante: si une case vide appartient à la ligne a, on y inscrit l'instruction $aS\varkappa_0$. Soit T' le programme obtenu. Si l'on convient que la machine correspondant à T' fonctionne comme la machine initiale à la seule différence qu'elle stoppe dès qu'elle se trouve dans l'état $\varkappa_0$, alors elle calculera la fonction f comme la machine initiale avec le programme T.

Considérons à l'instant t la bande de la machine T' et repérons sa zone de travail, c'est-à-dire l'ensemble composé des champs inspectés par l'œil et des champs dans lesquels a été inscrit le code initial de x. Le champ observé par l'œil à l'instant t divise la zone de travail

en deux parties: la partie gauche $\mathscr{L}_t$ et la partie droite $\mathscr{R}_t$ (voir fig. 33). La portion $\mathscr{L}_t$ est située à gauche du champ observé à l'instant t, $\mathscr{R}_t$ contient tous les autres champs de la zone de travail. Soient f_2 un entier naturel enregistré en binaire dans les champs de $\mathscr{L}_t$, f_1 un entier naturel enregistré en binaire dans les champs de $\mathscr{R}_t$ dans le sens droite-gauche. Il est manifeste que

$$f_1 = f_1(x', t), \quad f_2 = f_2(x', t),$$

où x' est un entier naturel dont la représentation binaire est confondue avec le code initial de x lu de droite à gauche.

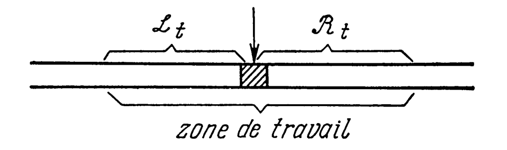


Fig. 33

Soit $f_3(x', t)$ le numéro de l'état à l'instant t, x' étant un entier naturel caractérisant l'enregistrement à l'instant initial.

A l'instant initial $t = 0$ on a $\mathscr{L}_t = \Lambda$ (vide), $\mathscr{R}_t$ est confondu avec l'ensemble des champs occupés par le code principal. Donc

$$\begin{cases} f_1(x', 0) = x', \\ f_2(x', 0) = 0, \\ f_3(x', 0) = 1. \end{cases}$$

Par ailleurs, il est évident que

$$\begin{cases} f_1(x', t+1) = \psi_1(f_1(x', t), f_2(x', t), f_3(x', t)), \\ f_2(x', t+1) = \psi_2(f_1(x', t), f_2(x', t), f_3(x', t)), \\ f_3(x', t+1) = \psi_3(f_1(x', t), f_2(x', t), f_3(x', t)). \end{cases}$$

En effet, sachant les nombres $f_1(x', t)$, $f_2(x', t)$ et $f_3(x', t)$ à l'instant t, on trouve le nombre observé par l'œil à l'instant t: ce sera le plus faible poids de la représentation binaire de $f_1(x', t)$, et l'état de la machine de numéro $f_3(x', t)$. Ces deux quantités nous permettent à l'aide du tableau T' de trouver la nouvelle valeur de ce champ, le nouvel état (le numéro de l'état) et la nature du déplacement et finalement les nombres $f_1(x', t+1)$, $f_2(x', t+1)$, $f_3(x', t+1)$. Ces conversions nous donnent les formules de ψ_1, ψ_2 et ψ_3. Explicitons-les. Désignons à cet effet par $T_1(z_1, z_2)$, $T_2(z_1, z_2)$ et $T_3(z_1, z_2)$ respectivement les fonctions définies par le tableau et délivrant d'après le symbole d'entrée et le numéro de l'état respectivement le nouveau symbole, le numéro du nouvel état et le numéro du déplacement (2 pour L, 1 pour S et 0 pour R). On admet que ces fonctions

sont nulles pour les autres valeurs de $E^{\aleph_0}$. Il est évident que T_1, T_2, $T_3 \in P_{\text{rp}}$, puisqu'elles prennent des valeurs non nulles en un nombre fini de points.

Désignons par $\chi(z)$ *) le plus faible poids de z et posons $T_3(\chi(u), w) = d$. On a alors

$$\psi_1(u, v, w) = \{(u \dot{-} \chi(u)) + T_1(\chi(u), w)\} \cdot d + \\ + (\overline{\text{Sg}}\, d)\left[\frac{u}{2}\right] + \chi(v) \cdot \text{Sg}(d \dot{-} 1),$$

$$\psi_2(u, v, w) = (2v + T_1(\chi(u), w))\, \overline{\text{Sg}}\, d + v \cdot \text{Sg}\, d \cdot \overline{\text{Sg}}\,(d \dot{-} 1) + \\ + \left[\frac{v}{2}\right] \text{Sg}(d \dot{-} 1),$$

$$\psi_3(u, v, w) = T_2(\chi(u), w).$$

D'où il résulte que ψ_1, ψ_2, $\psi_3 \in P_{\text{rp}}$.

Pour les fonctions f_1, f_2 et f_3 on a un schéma de récursion primitive simultanée. On peut donc affirmer que f_1, f_2, $f_3 \in P_{\text{rp}}$. Considérons maintenant

$$\mu_t \{f_3(x', t) = 0\}.$$

Cette fonction appartient à la classe P_{pr} et définit pour tout x' l'instant d'arrêt de la machine. Si l'on porte cette quantité dans f_1, on obtient

$$f_1(x', \mu_t \{f_3(x', t) = 0\}),$$

c'est-à-dire que si la machine stoppe, on obtient un entier naturel dont la représentation binaire est le code de $f(x)$.

Dans ce cas, si l'on tient compte du fait que

$$x' = \vartheta'(x_1, \ldots, x_n),$$

où $\vartheta'(x_1, \ldots, x_n) = \vartheta(x_n, \ldots, x_1)$, alors

$$f(x_1, \ldots, x_n) = \\ = \rho(f_1(\vartheta'(x_1, \ldots, x_n), \mu_t(f_3(\vartheta'(x_1, \ldots, x_n), t) = 0)) \dot{-} 1.$$

En posant

$$F_f(x_1, \ldots, x_n, y) = \rho(f_1(\vartheta'(x_1, \ldots, x_n), y) \dot{-} 1,$$
$$G_f(x_1, \ldots, x_n, y) = f_3(\vartheta'(x_1, \ldots, x_n), y),$$

on obtient ce qu'on voulait. C.q.f.d.

Une conséquence de ces théorèmes est le

Théorème 6. $P_{\text{cal}} = P_{\text{pr}}$.

Les deux derniers théorèmes nous disent que toute fonction partiellement récursive peut être représentée par des fonctions récur-

*) Il est évident que $\chi(z) \in P_{\text{rp}}$.

sives primitives sous forme d'une équation canonique définie par la représentation de Kleene.

Théorème 7. *Le système de fonctions* $\{0,\ S(x),\ I_1^1(x)\}$ *est complet dans* P_{cal} *pour les opérations* $\{S,\ \text{Rp},\ \mu\}$.

Théorème 8. *Le système de fonctions* $\{0,\ S(x)\}$ *est complet dans* P_{cal} *pour les opérations* $\{S,\ \text{Rp},\ \mu\}$.

Démonstration. La fonction $I_1^1(x)$ est définie au moyen de 0 et $S(x)$ à l'aide du schéma suivant:

$$\begin{cases} I_1^1(0) = 0, \\ I_1^1(x+1) = S(I_1^1(x)). \end{cases}$$

Théorème 9. *Le système fonctionnel* $(P'_{\text{cal}},\ S,\ \text{Rp},\ \mu)$ *contient l'analogue d'une fonction de Sheffer* *).

Démonstration. Soit une fonction $f(x_1,\ x_2)$ (voir tableau 40). Il est évident que $f(x_1,\ x_1) = S(x)$ et $f(x_1,\ S(x_1)) = 0(x_1)$. Comme

Tableau 40

x_1 \ x_2	0	1	2	3	...
0	1	0	0	0	...
1	0	2	0	0	...
2	0	0	3	0	...
3	0	0	0	4	...
...	...	...	...	...	...

dans le théorème précédent on obtient $I_1^2(x_1,\ x_2)$, $I_1^1(x_1) = I_1^2(x_1,\ x_1)$ et toutes les $I_m^n(x_1, \ldots, x_n)$. De là on peut construire sans peine la classe P'_{cal}.

*) Par P'_{cal} on désigne l'ensemble $P_{\text{cal}} \setminus E^{\aleph_0}$.

DEUXIÈME PARTIE

GRAPHES ET RÉSEAUX

La théorie des graphes et des réseaux peut être rattachée à la géométrie finie. L'intuition y joue un rôle essentiel dans la résolution des problèmes.

On se propose d'étudier ici des faits touchant à la réalisabilité d'une classe d'objets dans une autre, aux problèmes métriques relatifs aux graphes et réseaux et enfin aux particularités structurales de ces objets.

CHAPITRE 5

GRAPHES

§ 26. Réalisation dans l'espace euclidien. Isomorphisme

Dans la suite nous utiliserons souvent les notions d'« ensemble » et de « combinaison ». Le terme d'ensemble sera pris dans son sens usuel. Par « combinaison » on entendra une collection d'objets d'un ensemble dans laquelle un même objet peut apparaître plusieurs fois.

Définition. On appelle *graphe* Γ un ensemble $\mathfrak{M} = \{a_1, a_2, \ldots\}$ et une combinaison $\mathfrak{N}$ de couples d'objets (a_{i_k}, a_{j_k}) de $\mathfrak{M}$. Les objets de l'ensemble $\mathfrak{M}$ s'appellent *sommets du graphe*, ceux de la combinaison $\mathfrak{N}$, *arêtes du graphe.* On dira que l'arête (a_i, a_j) relie les sommets a_i et a_j.

Exemple 1. Soient $\mathfrak{M} = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$, $\mathfrak{N} = \{(a_1, a_2), (a_2, a_2), (a_4, a_5), (a_5, a_6), (a_5, a_6), (a_6, a_7), (a_5, a_7)\}$. Alors $\mathfrak{M}$ et $\mathfrak{N}$ définissent un graphe.

On dit que le graphe Γ est *fini* si l'ensemble $\mathfrak{M}$ et la combinaison $\mathfrak{N}$ sont composés d'un ensemble fini d'objets et de couples.

Soient a_i et a_j des sommets arbitraires du graphe Γ.

Définition. La séquence d'arêtes

$$A_{a_i a_j} = \{(a_{i_1}, a_{i_2}), (a_{i_2}, a_{i_3}), \ldots, (a_{i_{s-1}}, a_{i_s})\},$$

où $a_{i_1} = a_i$ et $a_{i_s} = a_j$ s'appelle *chaîne reliant les sommets a_i et a_j.* Si une arête appartient à une chaîne $A_{a_i a_j}$, on dira que *la chaîne $A_{a_i a_j}$ passe par* cette *arête*. De façon analogue, si un sommet a appartient à une arête de la chaîne $A_{a_i a_j}$, on dit que *la chaîne $A_{a_i a_j}$ passe par le sommet a.*

Définition. Une chaîne $A_{a_i a_j}$ est un *cycle*, si $a_i = a_j$. On appellera *boucle* un cycle (a_i, a_j).

Définition. On dit qu'un graphe Γ est *connexe* s'il existe une chaîne reliant deux quelconques de ses sommets a_i et a_j.

Il est immédiat de voir que le graphe de l'exemple 1 est fini, non connexe et contient des boucles. (Il est évident qu'un graphe connexe ne contient pas de sommets isolés, c'est-à-dire que chacun de ses sommets appartient au moins à l'une de ses arêtes.)

La notion de graphe que nous avons introduite est assez abstraite. Elle s'apparente à celle de complexe à une dimension en topologie. Pour étudier un graphe il nous faut l'interpréter de manière suggestive. Nous allons considérer à cet effet des figures d'une forme spéciale dans un espace euclidien. Chacune de ces figures $\mathfrak{z}$ est composée de sommets distincts b_1, b_2, . . . et de courbes (qui sont soit des arcs de cercle, soit des segments de droite) reliant chacune des couples de sommets (b_i, b_j) (le cas dégénéré $b_i = b_j$ n'est pas exclu). On convient qu'aucun point intérieur d'une courbe de la figure $\mathfrak{z}$ n'est sommet ou point intérieur d'une autre courbe.

Définition. On dit qu'une figure $\mathfrak{z}$ est une *réalisation géométrique d'un graphe* Γ si entre les sommets et les courbes de la figure $\mathfrak{z}$

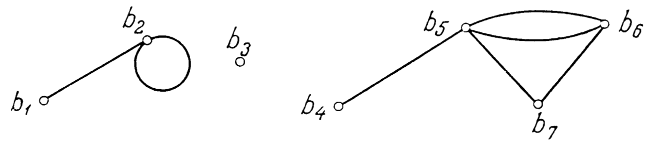


Fig. 34

d'une part et les sommets et les arêtes du graphe Γ d'autre part, il existe une correspondance biunivoque telle que si $(b_{n_i}, b_{n_j}) \leftrightarrow (a_i, a_j)$, alors $b_{n_i} \leftrightarrow a_i$, $b_{n_j} \leftrightarrow a_j$. (Les courbes et arêtes homologues relient les sommets homologues.)

Exemple 2. On s'assure sans peine que la figure $\mathfrak{z}$ de la figure 34 est la réalisation géométrique du graphe Γ de l'exemple 1.

Il se pose la question de savoir si tout graphe Γ peut être réalisé dans un espace euclidien, auquel cas il serait intéressant de savoir aussi s'il existe un nombre ρ tel que tout graphe admette une réalisation dans un espace euclidien de dimension ρ. Dans le dernier cas il serait souhaitable de connaître la valeur minimale de ρ. La réponse à toutes ces questions dans le cas de graphes finis est donnée par le théorème 1 et l'exemple 3. Le théorème 1 est une reformulation d'un théorème classique de topologie.

Théorème 1. *Tout graphe fini* Γ *peut être réalisé dans un espace euclidien à trois dimensions.*

Démonstration. Supposons qu'un graphe Γ contienne m sommets et h arêtes. Considérons une droite et traçons un faisceau de h plans passant par cette droite. Prenons m points $b_1, b_2, \ldots, b_m$ sur cette droite et associons-les respectivement aux sommets $a_1, a_2, \ldots, a_m$ du graphe. Associons un plan du faisceau à chaque arête de Γ. Soit (a_i, a_j) une arête du graphe Γ. Dans le plan correspondant à l'arête (a_i, a_j) relions les sommets b_i et b_j par un arc de cercle. En faisant la même construction pour les autres arêtes de Γ, on obtient une figure $\mathfrak{L}$ qui est visiblement la réalisation géométrique du graphe Γ.

On montre que ce théorème cesse d'être valable dans un espace euclidien de dimension inférieure à trois.

Exemple 3. La figure 35 représente deux graphes. Le premier est lié à la solution du célèbre problème des trois maisons et des trois

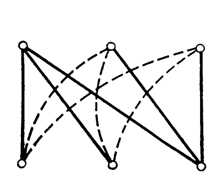
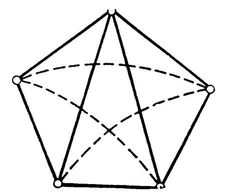



Fig. 35

puits *). Le second est un graphe complet **) à cinq sommets. En topologie on démontre que ces graphes ne peuvent pas être réalisés sur le plan. Nous omettrons la démonstration de ces faits.

Un résultat intéressant donnant des conditions de réalisabilité d'un graphe sur le plan a été obtenu par Pontriaguine [19] et indépendamment de lui, plus tard, par Kuratowski. Plus bas nous citons sans le prouver l'énoncé de ce fait (théorème 2).

*) Le problème des trois maisons et des trois puits s'énonce de la façon suivante: relier ces maisons et ces puits par des chemins ne se coupant pas.

**) C'est-à-dire le graphe contenant toutes les arêtes de la forme (a_i, a_j), $1 \leqslant i \leqslant j < m$.

Définition. Des graphes Γ et Γ' sont *isomorphes* s'il existe une correspondance biunivoque entre leurs sommets et leurs arêtes, telle que les arêtes correspondantes relient les sommets correspondants.

De ce point de vue, un graphe abstrait et sa réalisation géométrique sont des graphes isomorphes. En vertu du théorème 1, au lieu des graphes finis abstraits, on peut considérer leur réalisation. Autrement dit, on peut traiter les graphes comme des objets géométriques.

Introduisons l'opération de *subdivision d'une arête d'un graphe* Γ. Soient (a_i, a_j) une arête de Γ, a un objet n'appartenant pas à $\mathfrak{M}$.

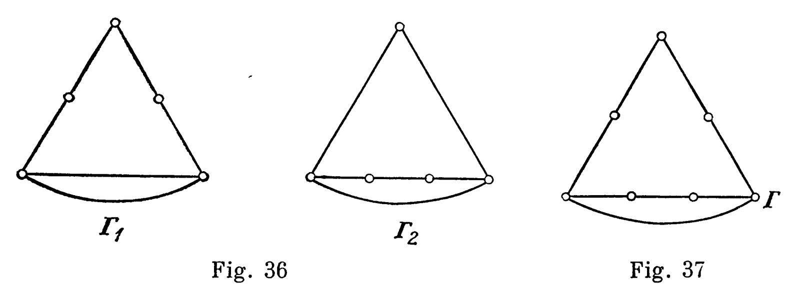


Fig. 36 Fig. 37

Subdiviser l'arête (a_i, a_j) de Γ revient à construire un graphe Γ' dont les sommets sont les objets de l'ensemble

$$\mathfrak{M}' = \mathfrak{M} \cup \{a\}$$

et dont les arêtes sont toutes celles du graphe Γ, à l'exception de (a_i, a_j), et les deux nouvelles arêtes (a_i, a), (a, a_j), c'est-à-dire que

$$\mathfrak{N}' = (\mathfrak{N} \setminus (a_i, a_j)) \cup \{(a_i, a), (a, a_j)\}.$$

Un graphe Γ_2 s'appelle *subdivision d'un graphe* Γ_1 s'il peut être déduit à partir de Γ_1 par un nombre fini de subdivisions des arêtes.

Définition. On dit que des graphes Γ_1 et Γ_2 sont *homéomorphes* s'ils possèdent des subdivisions isomorphes.

Exemple 4. La figure 36 représente deux graphes: Γ_1 et Γ_2. Ces graphes ne sont pas isomorphes, mais ils sont homéomorphes, car chacun d'eux peut être subdivisé au graphe Γ représenté sur la figure 37.

Définition. On dit qu'un graphe Γ' est un *sous-graphe* de Γ si ses sommets et arêtes appartiennent au graphe Γ.

Théorème 2 (critère de réalisabilité dans le plan.) *Pour qu'un graphe fini* Γ *soit réalisable dans un plan, il est nécessaire et suffisant qu'aucun de ses sous-graphes ne soit homéomorphe à aucun des graphes de la figure* 35.

§ 27. Estimation du nombre de graphes

Soit $\gamma(h)$ le nombre maximal de graphes deux à deux non isomorphes à h arêtes et sans sommets isolés. On se propose de majorer le nombre $\gamma(h)$.

Nous allons d'abord démontrer un théorème auxiliaire que nous utiliserons souvent dans la suite. Ce théorème relève de l'analyse combinatoire.

Supposons que H_n^m représente le nombre de toutes les combinaisons avec répétitions de n éléments m à m.

Exemple 5. Considérons un ensemble $\{a, b, c\}$ de trois éléments et formons toutes les combinaisons avec répétitions deux à deux.

$$(a, a),\ (b, b),\ (c, c),\ (a, b),\ (a, c),\ (b, c).$$

Donc, $H_3^2 = 6$.

Théorème 3. $H_n^m = C_{n+m-1}^m$.

Démonstration. Soit un ensemble d'objets $\{a_1, a_2, \ldots, a_n\}$.

Considérons une combinaison avec répétitions de ces éléments pris m à m. Supposons que cette combinaison est composée de s sortes d'objets $a_{i_1}, a_{i_2}, \ldots, a_{i_s}$ de multiplicités respectives m_1, $m_2, \ldots, m_s$, où $m_1 + m_2 + \ldots + m_s = m$. Nous écrirons cette combinaison sous la forme suivante:

$$a_{i_1}^{m_1} a_{i_2}^{m_2} \ldots a_{i_s}^{m_s}.$$

Etablissons une correspondance biunivoque entre les combinaisons avec répétitions des objets de l'ensemble $\{a_1, a_2, \ldots, a_m\}$ et les combinaisons ordinaires des objets de l'ensemble $\{a_1, \ldots, a_n, b_1, \ldots, b_{m-1}\}$, où $b_1, \ldots, b_{m-1}$ sont des symboles différents entre eux et de $a_1, \ldots, a_n$. Plus exactement: à la combinaison avec répétitions

$$a_{i_1}^{m_1} a_{i_2}^{m_2} \ldots a_{i_s}^{m_s}$$

associons la combinaison

$$a_{i_1} a_{i_2} \ldots a_{i_s} b_1 b_2 \ldots b_{m_1-1} b_{m_1+1} b_{m_1+2} \ldots$$
$$\ldots b_{m_1+m_2-1} b_{m_1+m_2+1} \ldots b_{m_1+\ldots+m_{s-1}+1} \ldots b_{m_1+\ldots+m_s-1}.$$

On voit que les objets de l'ensemble $\{b_1, b_2, \ldots, b_{m-1}\}$ figurent dans cette combinaison par groupes:

$$b_1 b_2 \ldots b_{m_1-1};\ b_{m_1+1},\ b_{m_1+2} \ldots b_{m_1+m_2-1};\ \ldots$$
$$\ldots b_{m_1+\ldots+m_{s-1}+1} \ldots b_{m_1+\ldots+m_s-1}.$$

Ces groupes sont au nombre de s et chacun d'eux contient respectivement $m_1 - 1$, $m_2 - 1$, . . ., $m_s - 1$ objets. Cette construction est correcte en vertu de l'identité

$$(m_1 - 1) + (m_2 - 1) + \ldots + (m_s - 1) + (s - 1) = m - 1.$$

Il s'ensuit encore que cette combinaison contient

$$s + (m_1 - 1) + (m_2 - 1) + \ldots + (m_s - 1) = m$$

objets. Donc, la combinaison construite contient des représentants de toutes les sortes d'objets de la combinaison avec répétitions a_{i_1}, a_{i_2}, . . ., a_{i_s} et les longueurs des groupes d'objets de l'ensemble $\{b_1, \ldots, b_{m-1}\}$ sont les codes des multiplicités d'entrées des objets a_{i_1}, . . ., a_{i_s} dans la combinaison initiale avec répétitions. De plus

la multiplicité de l'entrée de 1 est codée 0
» » » » 2 » » 1,
. . . .
» » » » m_i » » $m_i - 1$.

Si par exemple $n = 6$ et $m = 5$ et si l'on considère la combinaison avec répétitions $a_2^3 a_4^2$, alors d'après l'algorithme décrit à cette combinaison correspondra la combinaison $a_2 a_4 b_1 b_2 b_4$.

Montrons qu'à chaque combinaison des objets de l'ensemble $\{a_1, \ldots, a_n, b_1, \ldots, b_{m-1}\}$ pris m à m il correspond au sens indiqué plus haut une combinaison unique avec répétition (à partir de laquelle elle est déduite).

Il est évident que cette combinaison comprend s objets de l'ensemble $\{a_1, \ldots, a_n\}$ et $s \geqslant 1$: $a_{i_1} a_{i_2} \ldots a_{i_s}$. Les $m - s$ objets restant appartiennent à l'ensemble $\{b_1, \ldots, b_{m-1}\}$. Donc, $(m-1) - (m - s) = s - 1$ objets de l'ensemble $\{b_1, \ldots, b_{m-1}\}$ n'entrent pas dans la combinaison envisagée. Supposons que ce sont les objets de numéros

$$m_1,\ m_1 + m_2,\ \ldots,\ m_1 + m_2 + \ldots + m_{s-1}$$
$$(m_1,\ m_2,\ \ldots,\ m_{s-1} \geqslant 1).$$

Ces objets divisent l'ensemble $\{b_1, \ldots, b_{m-1}\}$ en s tranches dont certaines sont susceptibles d'être vides (dans le cas où le m_i correspondant est égal à 1). Les longueurs des tranches obtenues seront respectivement égales à

$$m_1 - 1,\ m_2 - 1,\ \ldots,\ m_s - 1.$$

En procédant au décodage, on obtient la combinaison avec répétition

$$a_{i_1}^{m_1} a_{i_2}^{m_2} \ldots a_{i_s}^{m_s}.$$

De cette correspondance il résulte immédiatement que

$$H_n^m = C_{n+m-1}^m.$$

Ce qui prouve le théorème.

Nous aurons besoin d'une inégalité pour la suite.

Lemme 1. $n! > \left(\frac{n}{e}\right)^n$.

Démonstration. On sait du cours d'analyse que $\left(1+\frac{1}{n}\right)^n < e$, d'où $n^n > \frac{(n+1)^n}{e}$. En s'appuyant sur ce fait on prouve par récurrence l'inégalité $n! > \left(\frac{n}{e}\right)^n$.

1. Base de la récurrence $n=1$, on a $1 > \frac{1}{e}$.

2. Passage de la récurrence

$$(n+1)! > (n+1)\left(\frac{n}{e}\right)^n > \frac{n+1}{e^n}\,\frac{(n+1)^n}{e} = \left(\frac{n+1}{e}\right)^{n+1}.$$

Ce qui prouve le lemme.

Théorème 4. $\gamma(h) > c_1 (c_2 h)^h$, *où* c_1 *et* c_2 *sont des constantes.*

Démonstration. Il est évident qu'un graphe de h arêtes ne possède pas plus de $2h$ sommets. Numérotons les sommets du graphe avec les entiers naturels 1, 2, . . . Il est évident que le nombre de toutes les sortes d'arêtes, c'est-à-dire de couples de sommets reliables par des arêtes, n'est pas supérieur à

$$r = H_{2h}^2 = C_{2h+1}^2 = h(2h+1).$$

Comme $\gamma(h)$ n'est pas supérieur au nombre maximal de graphes numérotés, deux à deux non équivalents et à h arêtes, et que ce nombre n'est pas supérieur au nombre de combinaisons avec répétitions de r éléments pris h à h, il vient

$$\gamma(h) \leqslant H_r^h = C_{r+h-1}^h \leqslant \frac{(r+h-1)^h}{h!} < \frac{(2h^2+2h)^h}{\left(\frac{h}{e}\right)^h} = \left(1+\frac{1}{h}\right)^h (2eh)^h < e\,(2eh)^h,$$

c.q.f.d.

Corollaire. *Le nombre maximal de graphes à h arêtes numérotés, deux à deux non isomorphes, n'est pas supérieur à* $c_1(c_2h)^h$.

Nous ne minorons pas $\gamma(h)$, ce qui nous prive de la possibilité de comprendre la qualité de la majoration obtenue.

CHAPITRE 6

RÉSEAUX

§ 28. Réseaux et leurs propriétés

Généralisons la notion de graphe.

Définition. L'ensemble $\mathfrak{M} = \{a_1, a_2, \ldots\}$ et la combinaison $\mathfrak{N} = \{E_0; E_1, E_2, \ldots\}$ dans laquelle chaque E_i est une combinaison d'éléments de $\mathfrak{M}$, i.e. $E_i = (a_{\nu_1(i)}, a_{\nu_2(i)}; \ldots)$, s'appelle *réseau* et se note $\mathfrak{M}\,(E_0; E_1, E_2, \ldots)$. Les objets de l'ensemble $\mathfrak{M}$ sont appelés *sommets*, ceux de E_0, *pôles du réseau* *).

Exemple 1. Soient

$$\mathfrak{M} = (1, 2, 3, 4, 5, 6, 7), \quad \mathfrak{N} = \{E_0; E_1, E_2, E_3, E_4, E_5\},$$

où

$$E_0 = (1, 2, 6), \; E_1 = (1, 3, 3, 4, 5),$$

$$E_2 = (4, 4, 4, 5, 6), \; E_3 = E_4 = (2, 4), \; E_5 = (2, 5, 6, 7).$$

Alors $\mathfrak{M}\,(E_0; E_1, E_2, E_3, E_4, E_5)$ est un réseau.

On dit que le réseau est *fini* si l'ensemble $\mathfrak{M}$ et la combinaison $\mathfrak{N}$ le sont. Ainsi, le réseau de l'exemple 1 est fini. Le réseau dans lequel $\mathfrak{M}$ ou $\mathfrak{N}$ est infini est dit *infini*. Les réseaux dénombrables, c'est-à-dire les réseaux dans lesquels $\mathfrak{M}$ et $\mathfrak{N}$ sont au plus dénombrables, sont un cas particulier de réseaux infinis.

On peut introduire la notion de réalisation géométrique d'un réseau fini ou dénombrable, comme pour le cas des graphes. Introduisons préalablement une notation. Désignons par $\langle E\rangle$ l'ensemble de tous les objets d'une combinaison E. Soit $\mathfrak{M}(E_0; E_1, \ldots)$

---

*) En littérature on trouve des notions proches: la notion d'organigramme et la notion d'hypergraphe. La notion d'organigramme est plus étroite que celle de réseau; la notion d'hypergraphe se distingue très peu de celles d'organigramme et de réseau. La notion d'organigramme est antérieure à celle de réseau, la notion d'hypergraphe, postérieure. Voir par exemple H a l l M., *Jr. Combinatorial Theory*, Toronto, 1967, Z y k o v A., *Hypergraphes*, UMN, 1974, t. 19, *vip.* 6 (180) (en russe).

un réseau. Divisons l'ensemble $\mathfrak{M}$ en trois parties disjointes:

$$\mathfrak{M}_1 = \langle E_0 \rangle \text{ l'ensemble des pôles,}$$

$$\mathfrak{M}_2 = \mathfrak{M} \setminus \bigcup_{i \geqslant 0} \langle E_i \rangle \text{ l'ensemble des sommets isolés distincts des pôles,}$$

$$\mathfrak{M}_3 \text{ les autres sommets.}$$

A chaque sommet des ensembles $\mathfrak{M}_1$ et $\mathfrak{M}_2$ associons bijectivement un point de l'espace. Affectons à ce point le symbole a_i du sommet correspondant de $\mathfrak{M}$. Il est évident qu'aux pôles seront associés des points affectés des symboles $a_{\nu_1(0)}$, $a_{\nu_2(0)}$, ... A chaque

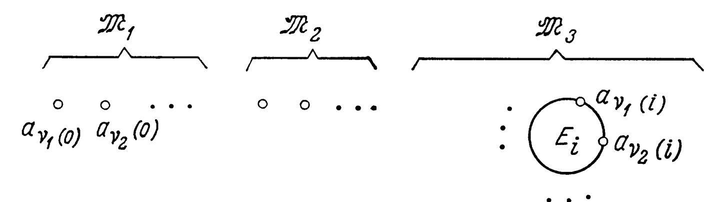


Fig. 38

combinaison $E_i = (a_{\nu_1(i)}, a_{\nu_2(i)}, \ldots)$ de $\mathfrak{N}$ $(i \geqslant 1)$ associons un cercle dans l'espace (si E_i contient un ou deux objets, on peut prendre un sommet ou un arc au lieu d'un cercle) sur la circonférence duquel sont choisis des sommets deux à deux distincts affectés des symboles $a_{\nu_1(i)}$, $a_{\nu_2(i)}$, ... de la combinaison E_i. Ceci étant, on exige que les cercles ne se coupent pas deux à deux et ne contiennent pas de sommets choisis antérieurement (fig. 38).

Les points qui sont affectés du même symbole a_i de $\mathfrak{M}$ sont reliés par une composante connexe A_i. On exige de plus que la composante connexe A_i ne possède en commun avec les sommets et les cercles construits antérieurement que les points affectés du symbole a_i et que les composantes connexes A_i et A_j $(i \neq j)$ ne possèdent pas de points communs. On appellera cette figure *réalisation géométrique du réseau initial.* Il est évident que les images des sommets a_i de $\mathfrak{M}_2$ du réseau $\mathfrak{M}$ $(E_0; E_1, \ldots)$ seront les sommets isolés a_i; les images des sommets a_i de $\mathfrak{M}_1$ et de $\mathfrak{M}_3$ seront soit des sommets isolés a_i (si le symbole a_i se présente une fois et dans une seule combinaison), soit la composante connexe A_i dans les autres cas; les images des combinaisons E_i $(i \geqslant 1)$ seront des cercles (resp. des sommets, des arcs).

On montre que tout réseau dénombrable admet une réalisation géométrique.

Exemple 2. La figure 39 représente la réalisation géométrique du réseau de l'exemple 1.

Cette figure rappelle le circuit d'un récepteur radio dont on a retiré tous les éléments: lampes, capacités, inductances, etc.

Définition. On dit que deux réseaux $\mathfrak{M}'$ $(E'_0;\ E'_1,\ E'_2,\ \ldots)$ et $\mathfrak{M}''$ $(E''_0;\ E''_1,\ E''_2,\ \ldots)$ sont *isomorphes* si l'on peut établir une correspondance biunivoque entre les objets des ensembles $\mathfrak{M}'$ et $\mathfrak{M}''$

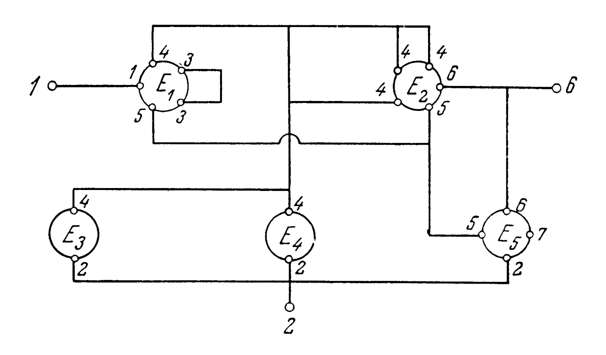


Fig. 39

et entre les objets des combinaisons $\mathfrak{N}'$ et $\mathfrak{N}''$ de telle sorte que:

1) les combinaisons correspondantes E' et E'' soient composées d'objets correspondants (compte tenu de leur multiplicité);

2) les combinaisons E'_0 et E''_0 se correspondent.

Il est évident qu'un réseau abstrait est isomorphe à sa réalisation géométrique. Comme nous nous intéresserons aux réseaux à l'isomorphisme près, on peut remplacer l'étude des réseaux abstraits par celle de leurs réalisations géométriques. De ce point de vue les réseaux sont des objets géométriques.

Considérons maintenant quelques classes de réseaux.

Il est évident que la classe des réseaux pour lesquels $E_0 = \Lambda$ et chaque combinaison E_i $(i \geqslant 1)$ est composée de deux objets de l'ensemble $\mathfrak{M}$, est confondue avec la classe des graphes.

Une autre classe de graphes est constituée par les arbres. On appelle *arbre* un graphe connexe fini de sommet distingué appelé racine, ne contenant pas de cycles.

Il est évident qu'un arbre est un réseau à un pôle, c'est-à-dire que $E_0 = (a)$.

Voici une autre définition d'un arbre équivalente à la première et reposant sur la récurrence. Il est plus commode de se servir de la définition sous la forme géométrique.

B a s e d e l a r é c u r r e n c e. La figure 40 représente un arbre de racine *a*.

P a s s a g e d e l a r é c u r r e n c e. Soient A (fig. 41, a) un arbre de racine a, B (fig. 41, b) un arbre de racine b.

Alors la figure C (fig. 42, a) obtenue à partir de A par « adjonction » d'une nouvelle arête à la racine a sera un arbre de racine c.

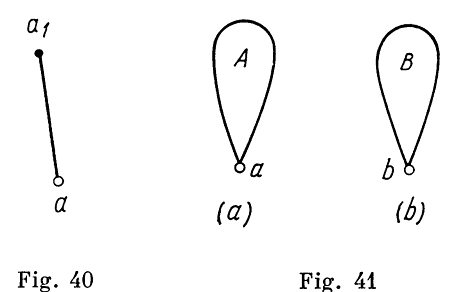


Fig. 40 Fig. 41

La figure D (fig. 42, b) obtenue à partir de A et de B par groupement des racines sera un arbre de racine c, où $c = a = b$.

On voit aussitôt que cette définition par récurrence de l'arbre peut être énoncée en termes de réseau abstrait.

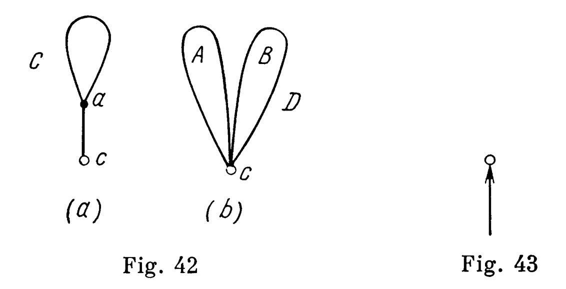


Fig. 42 Fig. 43

B a s e d e l a r é c u r r e n c e. Le réseau $\mathfrak{M}(E_0; E_1)$, où $\mathfrak{M} = \{a, a_1\}$, $E_0 = (a)$, $E_1 = (a, a_1)$ est un arbre de racine a.

P a s s a g e d e l a r é c u r r e n c e. Soient $A = \mathfrak{M}_1(E'_0; E'_1, \ldots)$ et $B = \mathfrak{M}_2(E''_0; E''_1, \ldots)$ des arbres de racines a et b, où $\mathfrak{M}_1 \cap \mathfrak{M}_2 = \Lambda$, $E'_0 = (a)$ et $E''_0 = (b)$. Alors le réseau $C = \mathfrak{M}(E_0; E, E'_1, \ldots)$ est un arbre de racine c si

$$\mathfrak{M} = \mathfrak{M}_1 \cup \{c\}, \ E_0 = (c), \ E = (a, c),$$

où c est un nouvel objet.

Le réseau $D = \mathfrak{M}'(E_0; \widetilde{E}'_1, \ldots, \widetilde{E}''_1, \ldots)$ est un arbre dont la racine est au sommet c, si

$$\mathfrak{M}' = (\mathfrak{M}_1 \setminus a) \cup (\mathfrak{M}_2 \setminus b) \cup \{c\}, \ E_0 = (c)$$

et la combinaison $\widetilde{E}'_i$ (resp. $\widetilde{E}''_i$) se déduit à partir de la combinaison E'_i (resp. E''_i) en remplaçant toutes les entrées du symbole a resp. (b) par c, où c est un nouvel objet.

La définition géométrique nous fournit un procédé de réalisation géométrique d'un arbre dans le plan. On appellera *immersion d'un*

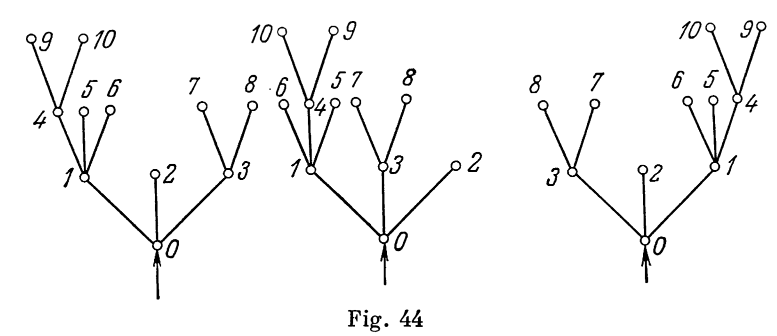


Fig. 44

arbre la réalisation géométrique plane dans laquelle les arêtes sont représentées par des segments de droite, et la racine, par un sommet avec une flèche (voir fig. 43).

Exemple 3. Soient

$$\mathfrak{M} = \{0,\ 1,\ 2,\ \ldots,\ 10\},\quad \mathfrak{N} = \{E_0;\ E_1,\ \ldots,\ E_{10}\},$$

où

$$E_0 = (0),\ E_1 = (0,\ 1),\ E_2 = (0,\ 2),$$

$$E_3 = (0,\ 3),\ E_4 = (1,\ 4),\ E_5 = (1,\ 5),$$

$$E_6 = (1,\ 6),\ E_7 = (3,\ 7),\ E_8 = (3,\ 8),\ E_9 = (4,\ 9),\ E_{10} = (4,\ 10).$$

Il est évident que $\mathfrak{M}$ (E_0; E_1, . . ., E_{10}) est un arbre. La figure 44 représente des immersions de cet arbre.

Considérons une immersion quelconque d'un arbre. On peut orienter les arêtes de cet arbre en se déplaçant de la racine vers les sommets. Chaque sommet (y compris la racine) est l'extrémité d'une arête et exception faite des terminaux, l'origine de plusieurs arêtes. Cette circonstance nous permet d'ordonner les arêtes incidentes à chaque sommet vers l'extérieur, par exemple dans l'ordre dans lequel on les rencontre en se déplaçant dans le sens des aiguilles d'une montre (voir fig. 45).

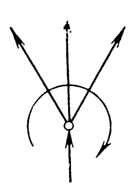


Fig. 45

Il est naturel d'admettre que deux immersions d'un arbre sont identiques si l'ordre de succession des arêtes incidentes aux sommets correspondants vers l'extérieur est le même. Donc, l'immersion d'un arbre est entièrement définie par l'ordre de succession des arêtes incidentes vers l'extérieur.

§ 29. Estimation du nombre de réseaux

Commençons par un problème simple. Désignons par $\delta(h)$ le nombre maximal d'arbres deux à deux non isomorphes à h arêtes, par $\delta^*(h)$ le nombre d'immersions des arbres de l'ensemble correspondant.

Théorème 1. $\delta(h) \leqslant \delta^*(h) < 4^h$.

Démonstration. Les immersions d'arbres non isomorphes étant distinctes, on a $\delta(h) \leqslant \delta^*(h)$.

A toute immersion d'un arbre de h arêtes on peut mettre en correspondance biunivoque un cortège de 0 et de 1 de longueur $2h$. Utilisons pour cela la définition par récurrence d'un arbre.

B a s e d e l a r é c u r r e n c e. Affectons le cortège 01 à l'immersion d'un arbre formé d'une seule arête. La longueur de ce cortège est égale à 2.

P a s s a g e d e l a r é c u r r e n c e. Supposons qu'aux immersions de deux arbres A et B de h_1 et h_2 arêtes respectivement on ait associé des cortèges α et β de longueur $2h_1$ et $2h_2$. Alors à l'immersion de l'arbre C, déduite à partir de celle de l'arbre A par adjonction d'une arête, associons le cortège $0\alpha 1$. Sa longueur est $2(h_1+1)$, c'est-à-dire le double du nombre d'arêtes de l'arbre A. A l'immersion de l'arbre D, obtenue à partir de celles des arbres A et B par réunion des racines, associons les cortèges $\alpha\beta$ ou $\beta\alpha$ selon leur ordre de succession. La longueur de chacun de ces cortèges est égale à $2(h_1+h_2)$, c'est-à-dire au double du nombre d'arêtes de l'arbre D.

On a

$$\delta^*(h) \leqslant C_{2h}^h < 2^{2h} = 4^h.$$

Ce qui prouve le théorème.

En comparant les majorations des nombres $\gamma(h)$ et $\delta(h)$ relatives au nombre de graphes et d'arbres, on voit que la dernière est essentiellement inférieure à la première (pour $h \to \infty$). Cette situation a son explication dans la structure topologique plus rigide des arbres.

Passons maintenant à la majoration du nombre de réseaux finis dans le cas général. Introduisons à cet effet quelques notations relatives au réseau

$$\mathfrak{M}(E_0;\ E_1,\ \ldots,\ E_h).$$

Supposons que e_i désigne le nombre d'objets (compte tenu de leurs répétitions) dans la combinaison E_i. La quantité e_i s'appelle *puissance de la combinaison.* Désignons par ε la puissance maximale d'une combinaison distincte de E_0, c'est-à-dire

$$\varepsilon = \max_{1 \leqslant i \leqslant h} e_i.$$

La quantité ε s'appelle *puissance du réseau.* Supposons ensuite que h_i ($1 \leqslant i \leqslant \varepsilon$) est le nombre de combinaisons de puissance i (sans compter la combinaison E_0). Le cortège $(h_1, h_2, \ldots, h_\varepsilon)$ s'appelle *structure exponentielle du réseau.* Il est manifeste que $\sum_{i=1}^{\varepsilon} h_i = h$. Introduisons enfin la quantité

$$\mu = \frac{1}{h} \sum_{i=1}^{\varepsilon} ih_i$$

que nous appellerons *puissance moyenne du réseau.*

Nous nous proposons d'étudier une classe de réseaux tels que

$$\mathfrak{M} = \bigcup_{i=0}^{h} \langle E_i \rangle.$$

Cette restriction signifie que ces réseaux ne possèdent pas de sommets « isolés » distincts des pôles et de sommets appartenant aux combinaisons.

Désignons par $S(e_0, h_1, \ldots, h_\varepsilon)$ le nombre maximal de réseaux deux à deux non isomorphes de cette classe, possédant e_0 pôles et une structure exponentielle donnée. Soit $S(e_0, \mu, \varepsilon, h)$ le nombre maximal de réseaux deux à deux non isomorphes de la même classe, possédant e_0 pôles, une puissance moyenne μ donnée, une puissance maximale ε et un nombre de combinaisons h (exceptée la combinaison E_0).

Ces quantités sont justiciables de majorations qui font le contenu du théorème 2 et de son corollaire.

Théorème 2 (O. Loupanov [9]).

$$S(e_0, h_1, \ldots, h_\varepsilon) \leqslant c(e_0, \mu, \varepsilon)^h \, h^{(\mu-1)h}.$$

Démonstration. Le nombre p de sommets dans les combinaisons $E_1, \ldots, E_h$ est visiblement égal à la quantité

$$\sum_{i=1}^{\varepsilon} ih_i = \mu h = p.$$

Les réseaux étant considérés à l'isomorphisme près, on peut admettre que les pôles sont

$$a_1, a_2, \ldots, a_{e_0}.$$

Evaluons le nombre de sortes de combinaisons de puissance i rencontrées dans ces réseaux. Il est clair que pour une quantité donnée p_i on a les relations

$$p_i = H_p^i = C_{p+i-1}^i \leqslant (p+i-1)^i \leqslant (2p)^i = (2\mu h)^i,$$

puisque

$$i \leqslant \varepsilon \leqslant \mu h = p.$$

On remarquera que

$$p_i \geqslant C_p^1 = p = \mu h \geqslant h_i.$$

Il est aisé de voir que le nombre de systèmes de h_i combinaisons de puissance i n'est pas supérieur à $H_{p_i}^{h_i}$. Pour $h_i \neq 0$ on a

$$H_{p_i}^{h_i} = C_{p_i+h_i-1}^{h_i} \leqslant \frac{(p_i+h_i-1)^{h_i}}{h_i!} \leqslant \frac{(2p_i)^{h_i}}{(h_i/e)^{h_i}} = \frac{(2ep_i)^{h_i}}{h_i^{h_i}}.$$

De là on déduit une majoration pour la quantité $S(e_0, h_1, \ldots, h_\varepsilon)$ que nous désignerons simplement par S:

$$S \leqslant (e_0+1) \prod_{\substack{i=1 \\ h_i \neq 0}}^{\varepsilon} H_{p_i}^{h_i}.$$

Le facteur e_0+1 traduit le fait que ou bien aucun pôle n'appartient à l'ensemble $\bigcup_{i=1}^{h} \langle E_i \rangle$, ou bien un pôle appartient à l'ensemble $\bigcup_{i=1}^{h} \langle E_i \rangle$ et ainsi de suite, ou bien enfin tous les e_0 pôles appartiennent à $\bigcup_{i=1}^{h} \langle E_i \rangle$. On a

$$S \leqslant (e_0+1) \prod_{\substack{i=1 \\ h_i \neq 0}}^{\varepsilon} \frac{(2e)^{h_i} (2\mu h)^{ih_i}}{h_i^{h_i}},$$

ou

$$\ln S \leqslant \ln (e_0+1) + \sum_{\substack{i=1 \\ h_i \neq 0}}^{\varepsilon} \ln \frac{(2e)^{h_i} (2\mu h)^{ih_i}}{h_i^{h_i}} =$$

$$= \ln (e_0+1) + h (\ln 2e + \mu \ln 2\mu) + \mu h \ln h - \sum_{\substack{i=1 \\ h_i \neq 0}}^{\varepsilon} h_i \ln h_i.$$

Soit $h_i = \xi_i h$. Il est évident que $\sum_{i=1}^{\varepsilon} \xi_i = 1$. Sous ces conditions on a l'inégalité (pour l'entropie)

$$-\sum_{i=1}^{\varepsilon} \xi_i \ln \xi_i \leqslant \ln \varepsilon.$$

(Pour $\xi = 0$ posons $\xi \ln \xi = 0$.) On obtient

$$\ln S \leqslant \ln (e_0+1) + h (\ln \varepsilon + \ln 2e + \mu \ln 2\mu) + (\mu-1) h \ln h.$$

D'où

$$S \leqslant (e_0+1)\,(2e\,\varepsilon\,(2\mu)^{\mu})^h h^{(\mu-1)h}.$$

Si l'on pose $c(e_0, \mu, \varepsilon) = (e_0+1)\,2e\varepsilon\,(2\mu)^{\mu}$, on a en définitive

$$S(e_0, h_1, \ldots, h_\varepsilon) \leqslant c(e_0, \mu, \varepsilon)^h h^{(\mu-1)h}.$$

Ce qui prouve le théorème.

La majoration obtenue dépend faiblement de la structure exponentielle $(h_1, \ldots, h_\varepsilon)$: elle ne contient que deux de ses caractéristiques, savoir μ et ε. Ceci permet de majorer sans peine $S(e_0, \mu, \varepsilon, h)$ pour toutes valeurs fixes de e_0, μ et ε. Pour cela il suffit de majorer le nombre de structures exponentielles $(h_1, \ldots, h_\varepsilon)$ de paramètres donnés μ, ε, h. Ce nombre peut être majoré par celui de solutions de l'équation $h_1 + h_2 + \ldots + h_\varepsilon = h$. Cela revient à disposer $\varepsilon - 1$ cloisons entre h unités. On se contentera de l'estimation grossière $(h+1)^{\varepsilon-1}$ (chaque cloison peut occuper $h+1$ positions). On obtient donc le

Corollaire.

$$S(e_0, \mu, \varepsilon, h) \leqslant c(e_0, \mu, \varepsilon)^h (h+1)^{\varepsilon-1} h^{(\mu-1)h} \leqslant$$
$$\leqslant c'(e_0, \mu, \varepsilon)^h h^{(\mu-1)h}.$$

Cette majoration englobe, comme un cas particulier, celle du nombre de graphes

$$\gamma(h) = S(0, 2, 2, h) \leqslant c^h h^h.$$

De là on déduit une majoration pour le nombre de graphes à deux sommets distingués et dont les sommets isolés ne sont pas des pôles (voir corollaire du théorème 4, chap. 5)

$$S(2, 2, 2, h) \leqslant c_1^h h^h.$$

§ 30. Réseaux bipolaires de combinaisons à deux objets

On se propose d'étudier ici une classe importante de réseaux finis bipolaires ($e_0 = 2$) constitués exclusivement de combinaisons à deux objets ($e_i = 2$, pour $i = 1, 2, \ldots, h$). Il est immédiat de voir que cette classe est confondue avec celle des graphes finis dans chacun desquels sont distingués deux sommets polaires. De tels réseaux

$$\mathfrak{M}(E_0; E_1, \ldots, E_h)$$

seront désignés par $\Gamma(a, b)$, où $E_0 = (a, b)$.

Pour les réseaux de même que pour les graphes on introduit la notion de chaîne reliant des sommets a^0, b^0. Si les sommets a^0 et b^0 coïncident respectivement avec des pôles a et b on se servira du terme « chaîne » sans en spécifier les sommets. Un réseau est *connexe* si le graphe correspondant l'est. Si le réseau $\Gamma(a, b)$ est connexe, alors

on peut indiquer une chaîne qui emprunte une arête quelconque de ce réseau. Remarquons aussi que pour les réseaux connexes

$$\mathfrak{M} \subseteq \bigcup_{i=1}^{h} \langle E_i \rangle.$$

Donc, un réseau connexe est entièrement défini par l'énumération de ses arêtes et l'indication de ses pôles.

Dans la suite de ce paragraphe on n'envisagera que des réseaux connexes.

Soient $\Gamma_1\ (a',\ b')$ et $\Gamma_2\ (a'',\ b'')$ deux réseaux connexes disjoints, c'est-à-dire

$$\Gamma_1(a',\ b') = \mathfrak{M}_1(E'_0;\ E'_1,\ \ldots,\ E'_{h'}),$$

$$\Gamma_2(a'',\ b'') = \mathfrak{M}_2(E''_0;\ E''_1,\ \ldots,\ E''_{h''}),$$

où $\mathfrak{M}_1 \cap \mathfrak{M}_2 = \Lambda$. Considérons une arête quelconque $E'_i = (a^0,\ b^0)$ du réseau $\Gamma_1\ (a',\ b')$. Par des raisonnements géométriques (voir

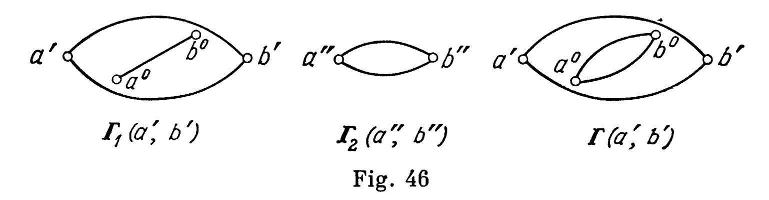


Fig. 46

fig. 46) on peut définir sans peine la substitution du réseau $\Gamma_2\ (a'',\ b'')$ à l'arête E'_i, substitution qui nous conduit au réseau $\Gamma\ (a',\ b')$.

Définition. On appelle *résultat de la substitution du réseau* $\Gamma_2\ (a'',\ b'')$ *à l'arête* $E'_i = (a^0,\ b^0)$ du réseau $\Gamma_1\ (a',\ b')$, chacun des réseaux $\Gamma'\ (a',\ b')$ et $\Gamma''\ (a',\ b')$:

$$\Gamma'(a',\ b') = \mathfrak{M}(E'_0;\ E'_1,\ \ldots,\ E'_{i-1},\ E^{\mathrm{III}}_1,\ \ldots,\ E^{\mathrm{III}}_{h''},\ E'_{i+1},\ \ldots,\ E'_{h'}),$$

$$\Gamma''(a',\ b') = \mathfrak{M}(E'_0;\ E'_1,\ \ldots,\ E'_{i-1},\ E^{\mathrm{IV}}_1,\ \ldots,\ E^{\mathrm{IV}}_{h''},\ E'_{i+1},\ \ldots,\ E'_{h'}),$$

où $\mathfrak{M} = \mathfrak{M}_1 \cup (\mathfrak{M}_2 \diagdown (a'',\ b''))$.

La combinaison E^{III}_j $(j = 1,\ \ldots,\ h'')$ se déduit à partir de E''_j par substitution de a^0 à a'' et de b^0 à b'', la combinaison E^{IV}_j $(j = 1,\ \ldots,\ h'')$ à partir de E''_j par substitution de b^0 à a'' et de a^0 à b''.

Définition. Le réseau $\Gamma\ (a,\ b)$ obtenu à partir de réseaux isomorphes à

$$\Gamma_1(a',\ b'),\ \ldots,\ \Gamma_m(a^{(m)},\ b^{(m)})$$

par application d'un nombre fini de substitutions s'appelle *superposition de ces réseaux.*

Exemple 4. Le réseau $\Gamma(a, b)$ de la figure 47 est la superposition des réseaux $\Gamma_1(a', b')$, $\Gamma_2(a'', b'')$, $\Gamma_3(a''', b''')$ (voir fig. 48).

En effet, considérons un réseau $\Gamma_4(a^{IV}, b^{IV})$ isomorphe au réseau $\Gamma_3(a''', b''')$ et substituons-le à une arête du réseau $\Gamma_3(a''', b''')$. Substituons le réseau obtenu à l'arête (c, d) du réseau $\Gamma_1(a', b')$. Finalement on obtient le réseau $\Gamma(a, b)$ en substituant le réseau $\Gamma_2(a'', b'')$ à l'arête (a', c) du réseau intermédiaire.

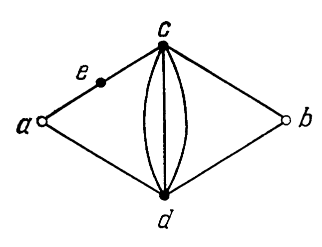


Fig. 47

Remarques. 1. Il est immédiat de voir que la substitution est une opération associative.

2. L'ensemble de tous les réseaux connexes $\{\Gamma(a, b)\}$ muni de la substitution est un système fonctionnel muni d'une opération.

Considérons dans le réseau $\Gamma(a, b)$ deux chaînes $A'_{a^0b^0}$ et $A''_{a^0b^0}$ reliant les sommets a^0 et b^0.

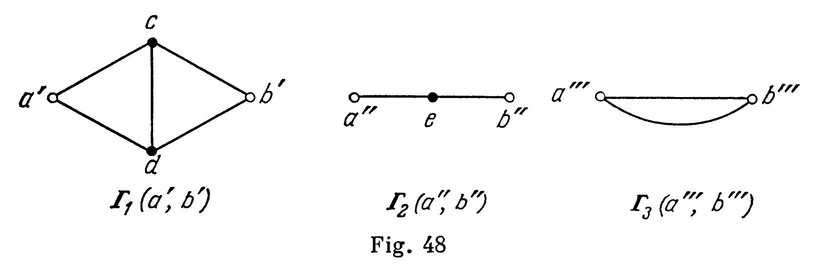


Fig. 48

Définition. La chaîne

$$A''_{a^0b^0}=\{(a^0, a_{i_2}), (a_{i_2}, a_{i_3}), \ldots, (a_{i_{s-1}}, b^0)\}$$

s'appelle *sous-chaîne de*

$$A'_{a^0b^0}=\{(a^0, a_{j_2}), (a_{j_2}, a_{j_3}), \ldots, (a_{j_{t-1}}, b^0)\},$$

si la suite d'arêtes

$$\{(a^0, a_{i_2}), (a_{i_2}, a_{i_3}), \ldots, (a_{i_{s-1}}, b^0)\}$$

s'obtient à partir de la suite d'arêtes

$$\{(a^0, a_{j_2}), (a_{j_2}, a_{j_3}), \ldots, (a_{j_{t-1}}, b^0)\}$$

par élimination d'un sous-ensemble d'arêtes. On appelle *sous-chaîne propre* $A''_{a^0b^0}$ de $A'_{a^0b^0}$ une sous-chaîne qui ne contient pas l'une au moins des arêtes de $A'_{a^0b^0}$.

Définition. Une chaîne $A_{a^0b^0}$ reliant deux points a^0 et b^0 d'un réseau $\Gamma(a, b)$ s'appelle *chaîne simple reliant ces points* si elle ne contient aucune sous-chaîne propre.

Remarque. Si les sommets a^0 et b^0 sont confondus avec les pôles a et b, on dira tout simplement « chaîne simple » au lieu de « chaîne simple reliant a et b ».

Exemple 5. Considérons le réseau $\Gamma(a, b)$ de la figure 49. Il est évident que $\{(a, c), (c, d), (d, c), (c, b)\}$ est une chaîne mais n'est pas une chaîne simple, car contient la sous-chaîne propre $\{(a, c), (c, b)\}$. La chaîne $\{(a, c), (c, b)\}$ est une chaîne simple.

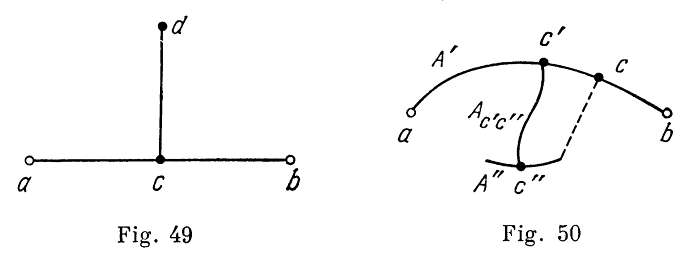


Fig. 49 Fig. 50

Il est immédiat de voir qu'un réseau contenant h ($h > 0$) arêtes possède une infinité de chaînes et un nombre fini de chaînes simples.

Introduisons encore une notion qui nous permettra de rétrécir la classe des réseaux étudiés.

Définition. Un réseau connexe $\Gamma(a, b)$ s'appelle *fortement connexe* si par chacune de ses arêtes il passe une chaîne simple.

Il est évident qu'un réseau connexe n'est pas nécessairement fortement connexe (voir exemple 5).

On démontre plus loin deux lemmes *). Le premier donne les conditions sous lesquelles il passe une chaîne simple par une arête donnée, et sert de base à la démonstration du second, lequel établit les conditions nécessaires et suffisantes de connexité forte.

Lemme 1. *Soit $\Gamma(a, b)$ un réseau (pas forcément connexe). Supposons que par les sommets c' et c'' ($c' \neq c''$) de Γ il passe des chaînes simples A' et A'' ($A' = A''$ n'est pas exclu). Si les sommets c' et c'' peuvent être reliés par une chaîne simple $A_{c'c''}$ ne possédant en commun avec A' et A'' que les sommets terminaux c' et c'', il existe alors une chaîne simple A dont $A_{c'c''}$ est une partie.*

Démonstration. Le lemme est évident si les sommets c' et c'' sont confondus avec les pôles du réseau Γ. On peut donc admettre que l'un au moins des sommets, par exemple c', est intérieur. Supposons pour fixer les idées que $c'' \neq b$. Désignons par c le premier sommet commun aux chaînes simples A' et A'', que l'on rencontre en suivant A'' du sommet c'' au pôle b. La figure 50 représente l'une

*) L'exposé suivant reprend les travaux de A. Kouznetsov (voir Troudy MIAN SSSR, tome 51, 1958) et de B. Trachtenbrot [27].

des deux situations possibles (si A'' est confondue avec A', alors $c = c''$). Désignons par A la chaîne obtenue à partir de la chaîne simple A' en substituant au tronçon $c'c$ la chaîne formée par $A_{c'c''}$ et la portion de A'' comprise entre les sommets c'' et c. Il est évident que A est la chaîne simple cherchée.

Considérons un sous-ensemble Γ' d'arêtes de réseau $\Gamma(a, b)$. Il est évident que Γ' définit un graphe. On dit qu'un sommet c de Γ' est un *sommet frontière* s'il est soit un pôle de $\Gamma(a, b)$, soit l'extrémité d'une arête de $\Gamma(a, b)$ n'appartenant pas à Γ'.

Définition. On dit qu'un sous-ensemble Γ' d'arêtes d'un réseau $\Gamma(a, b)$ est un *germe* s'il possède un seul sommet frontière.

Par exemple, le sous-ensemble d'arêtes $\{(c, d)\}$ de la figure 49 est un germe, puisqu'il possède un seul sommet frontière c; le sous-ensemble d'arêtes $\{(a, c), (c, b)\}$ n'en est pas un, car il possède trois sommets frontières: a, c, b.

Lemme 2. *Un réseau connexe $\Gamma(a, b)$ est fortement connexe si et seulement s'il ne contient pas de germes.*

Démonstration. Supposons qu'un réseau connexe $\Gamma(a, b)$ contient un germe Γ'. Désignons par c son sommet frontière. Considérons une arête quelconque appartenant à ce germe. Il est évident que

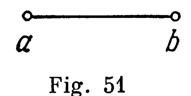


Fig. 51

toute chaîne empruntant cette arête doit passer au moins deux fois par le sommet c. Donc, par cette arête, il ne passe aucune chaîne simple. Par suite, le réseau $\Gamma(a, b)$ n'est pas fortement connexe.

Supposons maintenant que le réseau connexe $\Gamma(a, b)$ n'est pas fortement connexe. Montrons qu'il contient alors un germe. Comme $\Gamma(a, b)$ n'est pas fortement connexe, il existe des arêtes par lesquelles il ne passe aucune chaîne simple. Soit Γ' un sous-ensemble maximal connexe d'arêtes, doué de cette propriété. Le réseau $\Gamma(a, b)$ étant connexe, le graphe Γ' possède au moins un sommet frontière. Supposons que Γ' en possède au moins deux. Le graphe Γ' étant connexe, tout couple de sommets frontières peut être relié par une chaîne simple entièrement contenue dans Γ'. Supposons que c' et c'' sont des sommets frontières tels que la chaîne simple $A_{c'c''}$ ne contienne pas d'autres sommets frontières. Il est évident que par les sommets c' et c'' on peut mener des chaînes simples (reliant les pôles) A' et A''. Il est manifeste que la chaîne $A_{c'c''}$ ne possède en commun avec A' et A'' que les sommets terminaux c' et c''. En appliquant le lemme 1 aux chaînes A', A'' et $A_{c'c''}$ on obtient une chaîne dont $A_{c'c''}$ est

une partie. Or, ceci contredit la définition de Γ'. Donc Γ' possède un sommet frontière unique, autrement dit Γ' est un germe. C.q.f.d.

Dans la suite on n'envisagera que des réseaux fortement connexes. Soit le réseau $\Gamma_0(a, b) = \mathfrak{M}_0(E_0; E_1)$, où $\mathfrak{M} = \{a, b\}$, $E_0 = E_1 = (a, b)$ (voir fig. 51). Ce réseau s'appelle *réseau trivial.*

Définition. On dit qu'un réseau fortement connexe $\Gamma(a, b)$ est *décomposable* s'il existe des réseaux disjoints non triviaux $\Gamma_1(a, b)$ et $\Gamma_2(a', b')$ tels que $\Gamma(a, b)$ soit le résultat de la substitution du réseau $\Gamma_2(a', b')$ à une arête du réseau $\Gamma_1(a, b)$. Il est évident que le réseau de la figure 47 est décomposable.

Définition. Un réseau fortement connexe $\Gamma(a, b)$ qui n'est pas décomposable est dit *indécomposable.*

La figure 52 représente trois réseaux indécomposables non triviaux (les pôles sont marqués par des ronds).

On démontre que tout réseau fortement connexe indécomposable non trivial possédant six arêtes au plus est isomorphe à l'un des

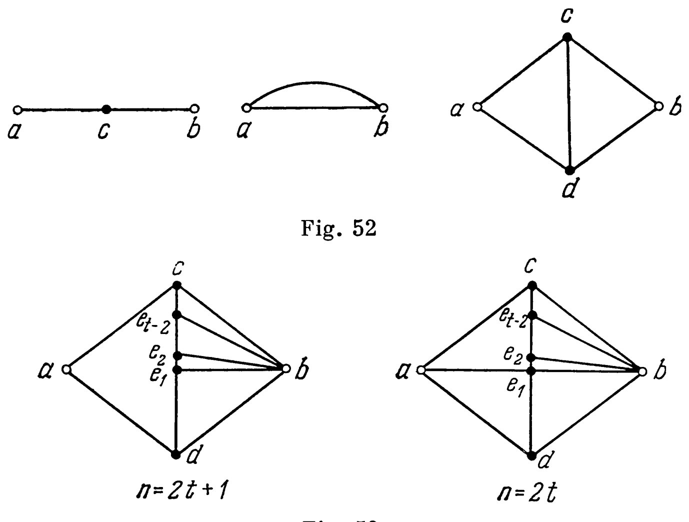


Fig. 52

Fig. 53

trois réseaux de la figure 52. Cela signifie qu'il n'existe pas de réseaux fortement connexes indécomposables à trois, quatre et six arêtes. Dans le même temps, pour tout $h \geqslant 7$ il existe des réseaux fortement connexes indécomposables à h arêtes *). Ceci ressort de la figure 53 qui représente des réseaux indécomposables à h ($h \geqslant 7$) arêtes.

*) Voir Kouznetsov A., Troudy MIAN SSSR, t. 51, 1958.

Dans la suite on se propose d'étudier la décomposabilité des réseaux. Etant donné qu'on s'occupera d'une décomposition d'une forme spéciale, on distinguera deux réseaux indécomposables élémentaires : la connexion en parallèle de deux arêtes $\Gamma_2^p(a, b)$ et la connexion en série de deux arêtes $\Gamma_2^s(a, b)$ (voir fig. 54). On désignera par H l'ensemble des autres réseaux indécomposables non triviaux et on appellera *H-réseau* tout réseau appartenant à H.

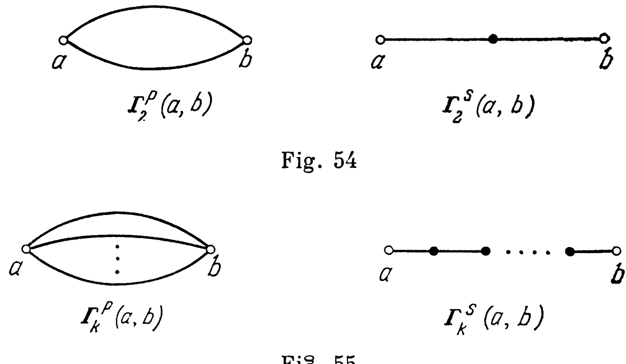


Fig. 54

Fig. 55

Deux ensembles infinis de réseaux sont rattachés aux réseaux $\Gamma_2^p(a, b)$ et $\Gamma_2^s(a, b)$. Il s'agit de

— l'ensemble P des réseaux $\Gamma_k^p(a, b)$, c'est-à-dire la connexion en parallèle de k arêtes ($k = 2, 3, \ldots$). Il est évident que pour $k > 2$ l'ensemble $\Gamma_k^p(a, b)$ est une superposition de réseaux $\Gamma_2^p(a, b)$ (voir fig. 55);

— l'ensemble S des réseaux $\Gamma_k^s(a, b)$, c'est-à-dire la connexion en série de k arêtes ($k = 2, 3, \ldots$). Il est évident que pour $k > 2$ l'ensemble $\Gamma_k^s(a, b)$ est la superposition des réseaux $\Gamma_2^s(a, b)$ (voir fig. 55).

Définition. Un réseau fortement connexe $\Gamma(a, b)$ *se décompose en deux tranches parallèles* si l'ensemble de toutes ses chaînes simples peut être divisé en deux classes non vides telles que deux chaînes simples quelconques appartenant à des classes distinctes ne possèdent pas de sommets intérieurs communs. Il est manifeste que chaque réseau $\Gamma_k^p(a, b)$ ($k \geqslant 2$) se décompose en deux tranches parallèles.

Définition. Soit c un sommet intérieur (c'est-à-dire différent d'un pôle) d'un réseau fortement connexe $\Gamma(a, b)$. On dit qu'un sommet c est *séparateur* si chaque chaîne simple passe par c. Il est évident que chaque sommet intérieur du réseau $\Gamma_k^s(a, b)$ est séparateur.

Soit c un sommet séparateur du réseau $\Gamma(a, b)$. Considérons dans chaque chaîne simple le tronçon qui va du sommet a au sommet c. Il est évident que l'ensemble de ces tronçons de chaînes simples engendre un réseau $\Gamma_1(a, c)$ dont les pôles sont les sommets a et c. De façon analogue, on obtient le réseau $\Gamma_2(c, b)$ en considérant les tronçons de chaînes simples compris entre les sommets c et b.

Lemme 3. *Soit c un sommet séparateur d'un réseau fortement connexe $\Gamma(a, b)$. Alors $\Gamma(a, b)$ est la superposition des réseaux $\Gamma_2^s(a, b)$, $\Gamma_1(a, c)$, $\Gamma_2(c, b)$ (connexion en série des réseaux $\Gamma_1(a, c)$ et $\Gamma_2(c, b)$).*

Démonstration. Elle découle du fait que les réseaux $\Gamma_1(a, c)$ et $\Gamma_2(c, b)$ ne possèdent pas de sommets intérieurs communs. En effet s'il en était autrement il existerait deux chaînes simples A_{ac}

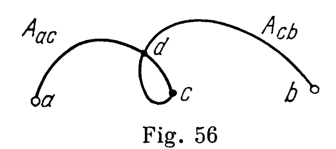


Fig. 56

et A_{cb} et, respectivement, deux réseaux $\Gamma_1(a, c)$ et $\Gamma_2(c, b)$, qui posséderaient un sommet intérieur commun (voir fig. 56). Désignons par d le premier sommet commun de ces chaînes simples, rencontré en parcourant A_{ac} à partir de a. Il est évident que la portion de A_{ac} comprise entre les sommets a et d et la portion de A_{cb} comprise entre les sommets d et b engendreraient une chaîne simple ne passant pas par c. Ce qui contredit le fait que le sommet c est séparateur. C.q.f.d.

Définition. Soit c un sommet d'un réseau $\Gamma(a, b)$. On appelle *étoile* (*de centre* c) l'ensemble de toutes les arêtes de $\Gamma(a, b)$ d'extrémités le sommet c. Si c est un pôle, l'étoile est dite *polaire*.

S'agissant des H-réseaux on a le lemme suivant:

Lemme 4. *Si $\Gamma(a, b)$ est un H-réseau, et c et d deux sommets intérieurs distincts (c'est-à-dire différents des pôles), alors il existe une chaîne simple passant par d et pas par c.*

Démonstration. Cette proposition découle d'un fait plus puissant: si de $\Gamma(a, b)$ on élimine l'étoile de centre c, on obtient un réseau qui est fortement connexe.

On distinguera trois cas.

a) L'élimination de l'étoile conduit à la décomposition du graphe en deux composantes connexes ne possédant pas de sommets communs (voir fig. 57). Il est évident alors que le sommet c sera séparateur et, en vertu du lemme 4, le réseau initial sera décomposable (l'une des composantes contiendra deux arêtes au moins, puisqu'un

H-réseau possède plus de quatre arêtes). Or, ceci est impossible.

b) L'élimination de l'étoile nous donne un réseau qui est connexe, mais pas fortement connexe. Le réseau obtenu n'étant pas fortement connexe, il possède un germe de point frontière d (voir fig. 58). Comme le réseau initial Γ (a, b) est fortement connexe, le germe doit

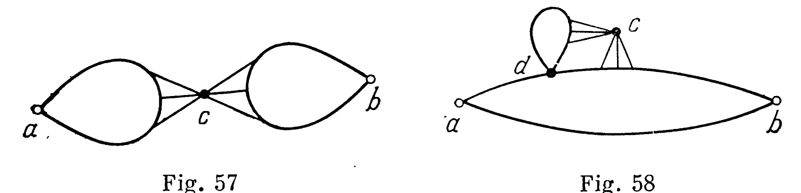


Fig. 57 Fig. 58

posséder des sommets frontières avec cette étoile, distincts de d. Dans ce cas, le germe forme avec une partie des arêtes de l'étoile un réseau bipolaire Γ' (d, c). Or, cela signifie que Γ (a, b) est décomposable, ce qui est absurde.

c) Reste le dernier cas, c'est-à-dire que l'élimination de l'étoile nous mène à un réseau fortement connexe. Ceci démontre le lemme.

Corollaire. *Un H-réseau ne possède pas de sommets séparateurs.*

Lemme 5. *Si* Γ (a, b) *est un H-réseau et* (a, c) *une arête appartenant à une étoile polaire, alors l'élimination de cette arête nous donne un réseau fortement connexe.*

Démonstration. Elle est identique à celle du lemme précédent.

a) L'élimination de l'arête nous donne un réseau qui n'est pas connexe. Il est évident que c sera alors un sommet séparateur du réseau Γ (a, b), ce qui contredit le corollaire du lemme 4.

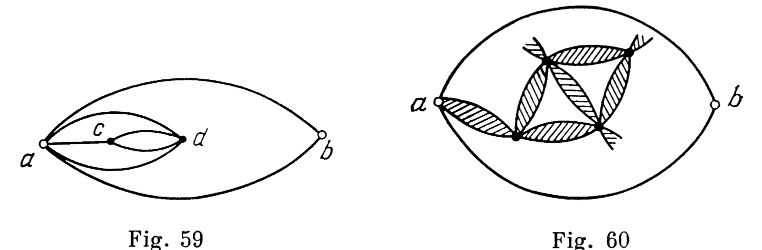


Fig. 59 Fig. 60

b) L'élimination de l'arête nous conduit à un réseau connexe Γ' (a, b), mais pas fortement connexe. Désignons par d le sommet frontière du germe du réseau Γ' (a, b) (voir fig. 59). Il est clair que ce germe forme avec l'arête (a, c) un réseau bipolaire Γ'' (a, d). Ce qui contredit l'indécomposabilité de Γ (a, b).

c) Reste le dernier cas: le réseau Γ' (a, b) est fortement connexe.

Considérons un réseau décomposable Γ (a, b). Supposons que

$\Gamma(a, b)$ est le résultat de la substitution des réseaux $\Gamma_1(a^{(1)}, b^{(1)}), \ldots$ $\ldots, \Gamma_h(a^{(h)}, b^{(h)})$, dont l'un au moins n'est pas trivial, respectivement aux arêtes $E_1, \ldots, E_h$ $(h \geqslant 2)$ du réseau $\Gamma_0(a, b) =$ $= \mathfrak{M}(E_0; E_1, \ldots, E_h)$.

La décomposition du réseau $\Gamma(a, b)$ en un réseau extérieur $\Gamma_0(a, b)$ et des réseaux intérieurs $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$ admet une interprétation géométrique simple: le réseau initial $\Gamma(a, b)$ est recouvert par les réseaux $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$ de telle sorte que deux réseaux intérieurs quelconques ne peuvent avoir en commun que leurs sommets polaires; la disposition de ces réseaux intérieurs est caractérisée par le réseau extérieur (voir fig. 60). Donc, chaque arête du réseau appartient exactement à un réseau intérieur; quant au sommet du réseau $\Gamma(a, b)$ soit c'est un pôle du réseau intérieur (par conséquent, le sommet du réseau extérieur), soit c'est le sommet intérieur d'exactement un réseau intérieur.

Remarque. Choisissons une chaîne simple dans le réseau $\Gamma_0(a, b)$. Soient

$$\Gamma_{i_1}(a^{(i_1)}, b^{(i_1)}), \ldots, \Gamma_{i_s}(a^{(i_s)}, b^{(i_s)})$$

les réseaux intérieurs correspondant aux arêtes de cette chaîne simple. Si l'on choisit une chaîne simple dans chacun de ces réseaux, on obtient une chaîne simple dans $\Gamma(a, b)$ (voir fig. 61).

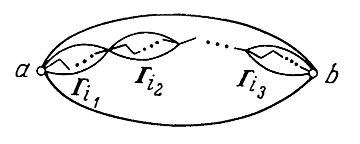


Fig. 61

Définition. Supposons que le réseau $\Gamma(a, b)$ se décompose en un réseau extérieur $\Gamma_0(a, b)$ et des réseaux intérieurs $\Gamma_1(a^{(1)}, b^{(1)}), \ldots$ $\ldots, \Gamma_h(a^{(h)}, b^{(h)})$. Alors:

a) si $\Gamma_0(a, b)$ est $\Gamma_h^p(a, b)$, le réseau $\Gamma(a, b)$ est dit *p-décomposable* et la décomposition correspondante, *p-décomposition*;

b) si $\Gamma_0(a, b)$ est $\Gamma_h^s(a, b)$, le réseau $\Gamma(a, b)$ est dit *s-décomposable* et la décomposition correspondante, *s-décomposition*;

c) si $\Gamma_0(a, b)$ est un H-réseau, le réseau $\Gamma(a, b)$ est dit *H-décomposable* et la décomposition correspondante, *H-décomposition.*

Lemme 6. *Si le réseau* $\Gamma(a, b)$ *est décomposable, on a exactement l'une des propositions suivantes:*

$\Gamma(a, b)$ *est p-décomposable;*

$\Gamma(a, b)$ *est s-décomposable;*

$\Gamma(a, b)$ *est H-décomposable.*

Démonstration. Il est immédiat de prouver que si $\Gamma(a, b)$ est décomposable, il admet soit une p-décomposition, soit une s-décomposition, soit une H-décomposition. Plus exactement, puisque $\Gamma(a, b)$ est décomposable, il se décompose en $\Gamma_0(a, b)$ (le réseau extérieur) et en $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$ (les réseaux intérieurs). Si le réseau extérieur est $\Gamma_h^p(a, b)$, ou $\Gamma_h^s(a, b)$, ou bien un H-réseau, alors on a l'une des décompositions indiquées. Si ce n'est pas le cas, alors $\Gamma_0(a, b)$ est décomposable et nous obtenons une autre décomposition du réseau $\Gamma(a, b)$ en un réseau $\Gamma_0'(a, b)$ (réseau extérieur) et en $\Gamma_1'(a_1^{(1)}, b_1^{(1)}), \ldots, \Gamma_{h'}(a_1^{(h')}, b_1^{(h')})$ (réseaux intérieurs), le réseau $\Gamma_0'(a, b)$ possédant moins d'arêtes que $\Gamma_0(a, b)$ $(h' < h)$. Si le réseau extérieur $\Gamma_0'(a, b)$ est $\Gamma_{h'}^p(a, b)$, ou $\Gamma_{h'}^s(a, b)$, ou bien un H-réseau, on obtient la décomposition cherchée. Dans le cas contraire le réseau $\Gamma_0'(a, b)$ est décomposable et l'on reprend de nouveau ce processus. Comme le réseau $\Gamma(a, b)$ possède un nombre fini d'arêtes, on aboutit en fin de compte à la décomposition voulue. Reste maintenant à montrer que le type de la décomposition se définit de façon unique.

Supposons que le réseau $\Gamma(a, b)$ admet une H-décomposition.

a) Il ne peut alors se décomposer en deux tranches parallèles. Si non, le réseau extérieur se décomposerait aussi en deux portions, autrement dit serait décomposable.

b) Un H-réseau ne peut posséder de sommet séparateur. Dans le cas contraire le sommet séparateur serait un sommet intérieur du réseau extérieur. Ce qui est impossible d'après le lemme 4.

Il est aussi clair que la décomposition du réseau en deux tranches parallèles et l'existence d'un sommet séparateur s'excluent mutuellement. Donc, un réseau bipolaire décomposable est doué exactement de l'une des propriétés suivantes: il se décompose en deux tranches parallèles, il possède un sommet séparateur, il admet une H-décomposition.

I. Un réseau décomposable se décompose en deux tranches parallèles si et seulement s'il est p-décomposable.

II. Un réseau décomposable possède un sommet séparateur si et seulement s'il est s-décomposable.

III. Un réseau décomposable ne se décompose pas en deux tranches parallèles et ne possède pas de sommet séparateur si et seulement s'il est H-décomposable.

Ceci achève la démonstration du lemme.

Définition. La p-décomposition du réseau $\Gamma(a, b)$ s'appelle *p-désintégration* si les réseaux intérieurs de la décomposition ne sont pas de la forme $\Gamma_2^p(a, b)$ et ne sont pas p-décomposables; la s-décomposition de $\Gamma(a, b)$ s'appelle *s-désintégration* si les réseaux intérieurs de la décomposition ne sont pas de la forme $\Gamma_2^s(a, b)$ et ne sont

pas s-décomposables ; la H-décomposition de $\Gamma(a, b)$ s'appelle *H-désintégration.*

On remarquera que les réseaux Γ_k^p et Γ_k^s $(k \geqslant 3)$ sont décomposables mais pas désintégrables.

Exemple 6. Considérons le réseau de la figure 62. Il est évident que la décomposition de ce réseau en le réseau $\Gamma_0'(a, b)$ et les réseaux $\Gamma_1'(a, c)$ et $\Gamma_2'(c, b)$ (voir fig. 63, *a*)) n'est pas une s-désintégration,

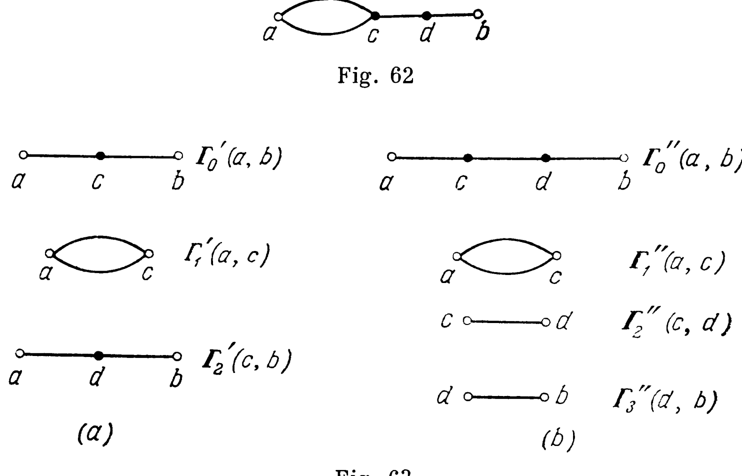


Fig. 62

Fig. 63

puisque le réseau $\Gamma_2'(c, b)$ est de la forme $\Gamma_2^s(c, b)$. Par contre, la s-décomposition en $\Gamma_0''(a, b)$ et en $\Gamma_1''(a, c)$, $\Gamma_2''(c, d)$ et $\Gamma_3''(d, b)$ (voir fig. 63, *b*)) est une s-désintégration.

Cet exemple montre que le réseau $\Gamma(a, b)$ admet plusieurs décompositions.

Définition. On dit qu'un sommet intérieur c d'un réseau $\Gamma(a, b)$ *dépend d'un sommet d* du même réseau si chaque chaîne simple qui passe par c passe également par d.

Définition. On dit que des sommets c et d d'un réseau $\Gamma(a, b)$ sont *équivalents* s'ils dépendent respectivement l'un de l'autre.

Définition. On dit qu'un sommet intérieur c d'un réseau $\Gamma(a, b)$ est *minimal* si, d étant un sommet intérieur quelconque de $\Gamma(a, b)$ ou bien c est équivalent à d ou bien c ne dépend pas de d.

Exemple 7. Considérons le réseau $\Gamma(a, b)$ de la figure 64. Le sommet f dépend de e ; le sommet e est équivalent à g ; les sommets c et d sont minimaux.

Des définitions il s'ensuit que les relations de dépendance et d'équivalence sont transitives, la deuxième étant de plus réflexive.

Lemme 7. *Si* $\Gamma(a, b)$ *est H-décomposable, alors l'ensemble des sommets minimaux est confondu avec celui des sommets intérieurs du réseau extérieur.*

Démonstration. Soit c un sommet minimal; montrons qu'il appartient à l'ensemble des sommets intérieurs du réseau extérieur. Supposons que ce n'est pas le cas. Alors c appartiendrait à un réseau intérieur. Il est évident qu'il dépendrait des sommets polaires de ce réseau. Comme le réseau extérieur est un H-réseau, l'un au moins de ces sommets polaires serait un sommet intérieur du réseau initial

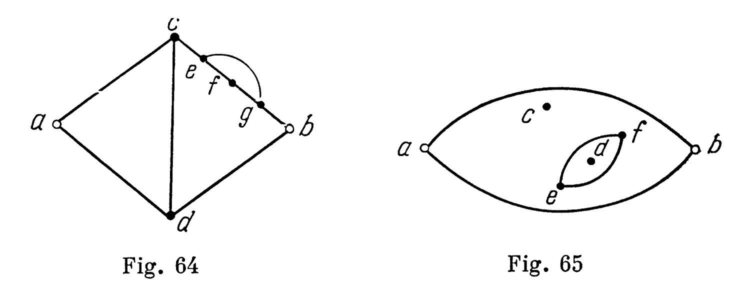


Fig. 64 Fig. 65

et c ne serait pas alors sommet minimal. Supposons maintenant que c est un sommet intérieur du réseau extérieur et montrons qu'il est minimal. Il suffit pour cela d'établir que c ne dépend d'aucun autre sommet intérieur d, c'est-à-dire qu'il existe une chaîne simple passant par c, mais pas par d. Il est évident que le sommet d est ou bien un sommet intérieur du réseau extérieur et l'on se sert alors du lemme du sommet intérieur d'un H-réseau, ou bien un sommet intérieur d'un réseau intérieur (voir fig. 65).

Désignons par e et f les pôles de ce réseau. On distinguera deux cas.

a) Le sommet e est confondu avec un pôle, a par exemple, le sommet f est confondu avec c. On applique alors le lemme 5. En tenant compte de la remarque de la page 173, on obtient une chaîne simple qui passe par c mais pas par d.

b) L'un des sommets, par exemple f, n'est confondu ni avec un pôle du réseau, ni avec c. On applique le lemme 4. Et toujours d'après la remarque de la page 173, on obtient une chaîne simple passant par c mais pas par d.

Théorème 3. *Si un réseau fortement connexe* $\Gamma(a, b)$ *est décomposable et n'est pas de la forme* Γ_k^p *et* Γ_k^s $(k \geqslant 3)$, *il admet alors une désintégration unique.*

Démonstration. Nous avons déjà vu que le type de désintégration est unique. Il reste donc à prouver que cette désintégration est unique dans le cadre de ce type.

P r e m i e r c a s. Le réseau $\Gamma(a, b)$ se décompose en deux tranches parallèles. Divisons toutes les chaînes simples en classes. Deux chaînes simples A' et A'' appartiendront à la même classe si et seulement si elles contiennent des sommets intérieurs (respectivement c' et c'') susceptibles d'être reliés par une chaîne simple $A_{c'c''}$ ne passant pas par les pôles (il s'ensuit que tout couple de sommets intérieurs de ces chaînes peut être relié par une chaîne simple ne passant pas par les pôles). Cette définition est correcte, puisque si les sommets intérieurs des chaînes A' et A'' pouvaient être réunis par des chaînes simples ne passant pas par les pôles, et les sommets intérieurs des chaînes A'' et A''', par des chaînes simples ne passant pas par les pôles, alors les sommets intérieurs de A' et A''' pourraient également être reliés par des chaînes simples ne passant pas par les pôles. On obtient donc une partition en les classes

$$\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_h,$$

et cette partition est unique. La partition obtenue engendre une p-désintégration en les réseaux

$$\Gamma_h^p(a, b), \Gamma_1(a, b), \ldots, \Gamma_h(a, b)$$

(les réseaux $\Gamma_i(a, b)$ $(i = 1, \ldots, h)$ ne sont pas des réseaux $\Gamma_2^p(a, b)$ et ne sont pas p-décomposables; de plus l'un d'eux au moins n'est pas trivial, puisque $\Gamma(a, b) \neq \Gamma_h^p(a, b)$).

D e u x i è m e c a s. Le réseau $\Gamma(a, b)$ possède des sommets séparateurs $c_1, \ldots, c_{h-1}$. Il est immédiat de voir qu'en cheminant sur une chaîne simple quelconque du pôle a vers le pôle b, on rencontre les sommets séparateurs dans le même ordre (par exemple $c_1, \ldots, c_{h-1}$). Soient $\Gamma_1(a, c_1)$, $\Gamma_2(c_1, c_2)$, $\ldots$, $\Gamma_h(c_{h-1}, b)$ les réseaux formés par les portions de chaînes simples comprises entre les sommets correspondants. Comme pour le lemme 3, on montre aisément que ces réseaux ne possèdent pas de sommets communs différents des pôles. On obtient une s-désintégration du réseau $\Gamma(a, b)$ en le réseau $\Gamma_h^s(a, b)$ et les réseaux $\Gamma_1(a_1, c_1)$, $\Gamma_2(c_1, c_2)$, $\ldots$ $\ldots$, $\Gamma_h(c_{h-1}, b)$, puisque $\Gamma(a, b) \neq \Gamma_h^s(a, b)$. Par suite, l'un au moins des réseaux intérieurs n'est pas trivial. L'unicité découle de la construction.

T r o i s i è m e c a s. Le réseau $\Gamma(a, b)$ admet une H-décomposition qui est unique en vertu du lemme 7.

En effet, les sommets minimaux définissent tous les sommets intérieurs du réseau extérieur; un couple de sommets minimaux ou

de pôles c et d définit une arête du réseau extérieur si et seulement s'il existe une chaîne simple reliant c et d et ne passant pas par d'autres sommets minimaux ou pôles. Ce qui prouve le théorème.

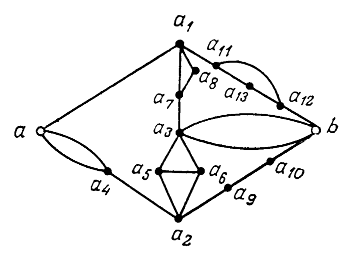


Fig. 66

Considérons un réseau $\Gamma(a, b)$. En le désintégrant (si cela est possible) on obtient des réseaux présentant un nombre moindre d'arêtes. Désintégrons ensuite les réseaux intérieurs qui peuvent l'être. En poursuivant cette procédure on arrive finalement à des réseaux qui sont soit triviaux, soit indécomposables, soit de la forme Γ_k^p, Γ_k^s $(k \geqslant 3)$. Le système de tous les réseaux non triviaux obtenu par désintégration de $\Gamma(a, b)$ s'appelle *désintégration canonique du réseau* $\Gamma(a, b)$. On a donc prouvé le

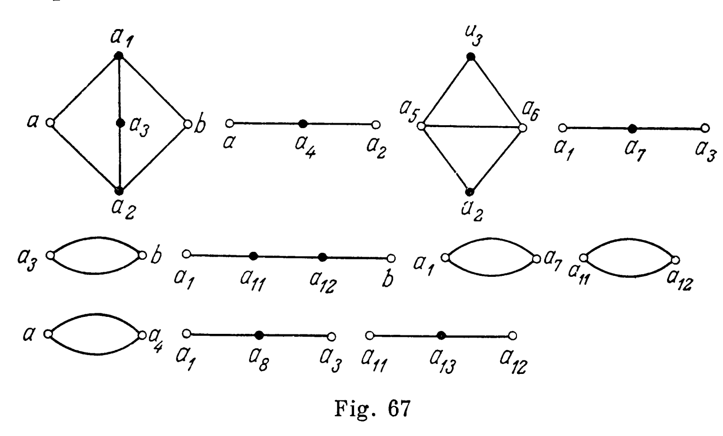


Fig. 67

Théorème 4. *Tout réseau fortement connexe non trivial* $\Gamma(a, b)$ *qui est désintégrable admet une désintégration canonique unique.*

Exemple 8. Le réseau $\Gamma(a, b)$ de la figure 66 se décompose en réseaux par désintégration canonique (voir fig. 67).

§ 31. π-réseaux

La classe des π-réseaux constitue une importante sous-classe de réseaux bipolaires de combinaisons à deux objets.

Définition. On appelle *π-réseau* la superposition des réseaux $\Gamma_2^p(a, b)$ et $\Gamma_2^s(a, b)$.

Voici une autre définition équivalente: un réseau bipolaire, fortement connexe, de combinaisons à deux objets est un π-réseau si sa désintégration canonique est composée de réseaux de la forme $\Gamma_h^p(a, b)$ et $\Gamma_h^s(a, b)$.

Exemple 9. Le réseau $\Gamma(a, b)$ de la figure 68 est un π-réseau.

A chaque π-réseau on peut associer un ensemble d'immersions d'un arbre dont les sommets non terminaux sont affectés des symboles p et s. Deux cas sont possibles:

1) $\Gamma(a, b) = \Gamma_h^\sigma(a, b)$, où $\sigma = p$ ou $\sigma = s$. Associons au réseau $\Gamma_n(a, b)$ un faisceau de h ($h \geqslant 2$) arêtes *) dont la racine est affectée du symbole σ (fig. 69).

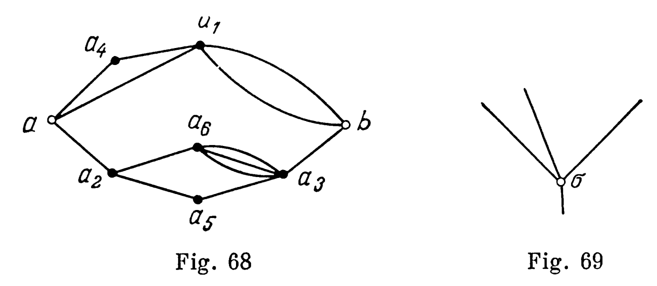


Fig. 68 Fig. 69

2) $\Gamma(a, b)$ se désintègre en le réseau $\Gamma_h^\sigma(a, b)$ et les réseaux $\Gamma_1(a^{(1)}, b^{(1)}), \ldots, \Gamma_h(a^{(h)}, b^{(h)})$ ($\sigma = p$ ou $\sigma = s$). Menons à partir de la racine affectée du symbole σ les h ($h \geqslant 2$) arêtes **) qui correspondent aux réseaux intérieurs (fig. 69).

Aux extrémités des arêtes auxquelles correspondent des réseaux non triviaux associons un symbole différent de σ (et que nous désignerons par $\bar{\sigma}$ ***)). Ensuite pour chaque réseau non trivial $\Gamma_i(a^{(i)}, b^{(i)})$ on applique soit le cas 1, soit le cas 2 et on construit des faisceaux aux sommets $\bar{\sigma}$, et ainsi de suite (fig. 70). Dans l'immersion construite chaque faisceau contient deux arêtes au moins. Donc, à tout π-réseau correspond un ensemble d'immersions d'arbres. Aux π-réseaux non isomorphes correspondent des ensembles disjoints d'immersions d'arbres. Donc, le nombre de π-réseaux n'est pas supérieur à celui des immersions d'arbres.

Considérons les immersions d'arbre correspondant au π-réseau de l'exemple 9 (fig. 71). Il est immédiat de voir que l'immersion associée à un π-réseau de h arêtes possède h sommets terminaux.

*) L'ordre des arêtes est arbitraire pour $\sigma = p$ et correspond à celui des arêtes du réseau $\Gamma_h^s(a, b)$ pour $\sigma = s$.

**) Idem.

***) Où $\bar{\sigma} = s$ pour $\sigma = p$, et $\bar{\sigma} = p$ pour $\sigma = s$.

Ainsi, l'étude des π-réseaux peut être ramenée à celle des immersions d'arbres de forme spéciale. Montrons que ce lien permet d'étendre aux π-réseaux certains faits concernant les arbres.

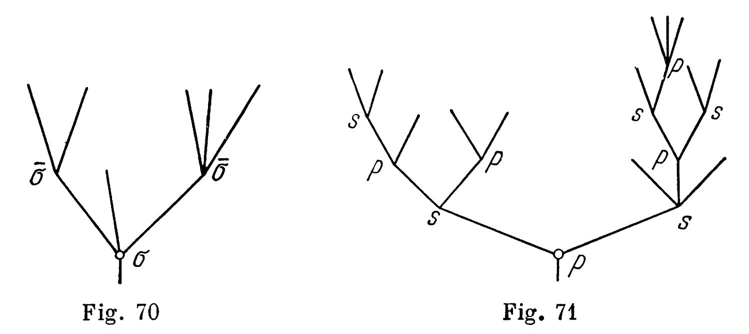


Fig. 70 Fig. 71

Théorème 5. *Soit π (h) le nombre maximal de π-réseaux à h arêtes deux à deux non isomorphes. Alors* $\pi(h) \leqslant 4^{2h}$.

Démonstration. Il est évident que le nombre cherché n'est pas supérieur à celui des immersions des arbres à h sommets terminaux, dont chaque faisceau d'arêtes incidentes vers l'extérieur contient deux arêtes au moins. Désignons par l le nombre d'arêtes d'un arbre de cette classe. On montre par récurrence que $l \leqslant 2h - 2$ pour $h \geqslant 2$.

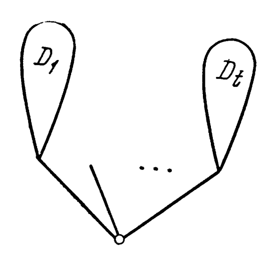


Fig. 72

B a s e d e l a r é c u r r e n c e. Si le π-réseau contient deux arêtes ($h = 2$), il est évident que l'arbre correspondant comporte deux arêtes terminales, c'est-à-dire que $l = 2$ et l'inégalité a lieu.

P a s s a g e d e l a r é c u r r e n c e. Supposons que l'inégalité est réalisée pour les arbres correspondant aux π-réseaux comportant h arêtes au plus. Considérons un π-réseau $\Gamma(a, b)$ à $h + 1$ arêtes et l'arbre correspondant (fig. 72). Si $\Gamma(a, b) = \Gamma_h^\sigma(a, b)$, alors $l = h > 2$ et l'inégalité a lieu de toute évidence. Si $\Gamma(a, b)$ est désintégrable, le nombre m d'arêtes incidentes à la racine de l'arbre vers l'extérieur est égal à celui des arêtes du réseau extérieur de la désintégration de $\Gamma(a, b)$ et, par hypothèse, $m \geqslant 2$. Les arbres $D_1, \ldots, D_t$ correspondent aux réseaux intérieurs non triviaux de cette désintégration ($t \leqslant m$). Désignons par l_i et h_i ($i = 1, \ldots, t$) le nombre d'arêtes et le nombre de sommets terminaux de l'arbre D_i. D'après l'hypothèse de la récurrence, $l_i \leqslant 2h_i - 2$ ($i = 1, \ldots, t$) et de plus il est évident que $\sum_{i=1}^{t} l_i + m = l$, $\sum_{i=1}^{t} h_i + (m - t) = h$.

On a

$$l=\sum_{i=1}^{t} l_i+m\leqslant 2\sum_{i=1}^{t} h_i-2t+m=$$

$$=2\left(\sum_{i=1}^{t} h_i+(m-t)\right)-m=2h-m\leqslant 2h-2,$$

puisque $m\geqslant 2$.

Pour majorer $\pi(h)$ on remarquera que dans chaque immersion d'un arbre de cette classe, les symboles p et s peuvent être distribués de deux manières. De ce fait, en se servant de la majoration pour le nombre d'immersions des arbres dont le nombre des arêtes est donné, on obtient

$$\pi(h)\leqslant 2\cdot 4^{2h-2}<4^{2h}.$$

TROISIÈME PARTIE

THÉORIE DU CODAGE

Les problèmes de codage jouent un rôle fondamental en mathématiques. Le codage permet de ramener l'étude de certains êtres à celle d'autres. On connaît bien la place tenue par la représentation des nombres dans le système décimal. La méthode des coordonnées a beaucoup contribué à l'essor des mathématiques en ce sens qu'elle a permis de coder des êtres géométriques à l'aide d'expressions

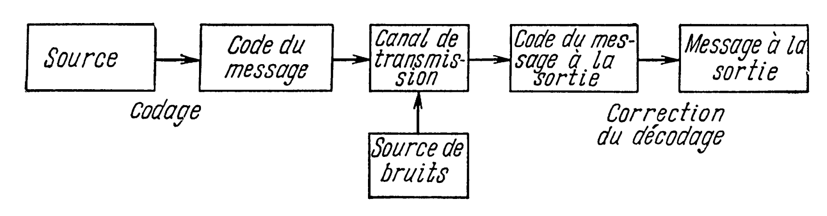


Fig. 73

analytiques. Cependant les méthodes de codage n'ont été qu'un outil auxiliaire et n'ont pas fait l'objet d'études. Les systèmes commandés ont donné une dimension nouvelle aux codes. Des études systématiques se sont avérées nécessaires dans le domaine de la théorie du codage. Les problèmes essentiels de la théorie du codage peuvent être illustrés sur un exemple emprunté au domaine des liaisons, exemple qui est schématisé sur la figure 73.

Soit donné un alphabet $\mathfrak{A} = \{a_1, \ldots, a_r\}$ composé d'un nombre fini de lettres. On appellera *mot* une séquence finie de symboles de $\mathfrak{A}$:

$$A = a_{i_1} a_{i_2} \ldots a_{i_n}.$$

Soient $S(\mathfrak{A})$ l'ensemble de tous les mots formés à l'aide de l'alphabet $\mathfrak{A}$, S' un sous-ensemble de S. On appelle *source d'information* l'être engendrant des mots de S', et *messages*, des mots de S'. La source d'information peut être un appareil, l'homme, etc. Lorsqu'on a à résoudre des problèmes de théorie du codage, on fait appel en général à une information supplémentaire sur la source, qui consiste en une

description de cette dernière. Il existe plusieurs méthodes de description des sources d'informations:

a) la *description ensembliste* se traduit par la fixation des caractéristiques cardinales, par exemple, S' est l'ensemble de tous les mots de longueur donnée m;

b) la *description statistique* revient à donner des caractéristiques probabilistes de S', par exemple, $S' = S$, et à donner les probabilités $p_1, \ldots, p_r$ d'apparition des lettres $a_1, \ldots, a_r$ $\left(\sum_{i=1}^{r} p_i = 1\right)$;

c) la *description logique* nous donne l'ensemble S' comme un « langage ». Elle caractérise les méthodes de construction de l'ensemble S', par exemple, S' peut être engendré par un automate.

Soit donné un alphabet $\mathfrak{B}$ où

$$\mathfrak{B} = \{b_1, \ldots, b_q\}.$$

Désignons par B un mot et par $S(\mathfrak{B})$ l'ensemble de tous les mots formés à l'aide de l'alphabet $\mathfrak{B}$.

Soit donnée une application F qui à chaque mot $A \in S'(\mathfrak{A})$ associe un mot

$$B = F(A), \quad B \in S(\mathfrak{B}).$$

Le mot B sera appelé *code du message* A et le passage du mot A à son code, *codage.*

En théorie du codage l'application F est définie par un algorithme.

Exemple 1. a) *Codage alphabétique.* Considérons la correspondance entre les lettres de l'alphabet $\mathfrak{A}$ et certains mots de $S(\mathfrak{B})$:

$$\begin{aligned} &a_1 - B_1,\\ &a_2 - B_2, \qquad\qquad (\Sigma)\\ &\cdot\;\cdot\;\cdot\;\cdot\;\cdot\\ &a_r - B_r. \end{aligned}$$

Cette correspondance s'appelle *schéma* et se note Σ. Elle définit un codage alphabétique de la manière suivante: à chaque mot $A = a_{i_1} \ldots a_{i_n}$ de $S'(\mathfrak{A}) = S(\mathfrak{A})$ est associé un mot $B = B_{i_1} \ldots$ $\ldots B_{i_n}$ dit *code du mot* A. Les mots $B_1, \ldots, B_r$ s'appellent *codes élémentaires.*

b) *Codage uniforme.* Soit $\{A_1, \ldots, A_s\}$ un sous-ensemble de mots deux à deux distincts de $S(\mathfrak{A})$, de même longueur m. Il est évident que chaque mot A décomposable sous la forme

$$A = A_{i_1} \ldots A_{i_n},$$

l'est de façon unique. Supposons par ailleurs que $S'(\mathfrak{A})$ est un sous-ensemble de mots de $S(\mathfrak{A})$ admettant une décomposition de la

forme indiquée plus haut. Considérons le schéma

$$\begin{array}{c} A_1 - B_1, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ A_s - B_s, \end{array} \qquad (\Sigma)$$

où les codes élémentaires B_i sont de même longueur.

Le schéma Σ définit un codage uniforme de la manière suivante: à chaque mot $A = A_{i_1} \ldots A_{in}$ de $S'(\mathfrak{A})$ est associé un mot $B = B_{i_1} \ldots B_{in}$ appelé code de A.

Le choix des codes est conditionné par plusieurs facteurs, notamment par

— la commodité de transmission (par exemple, le code binaire est d'un usage technique aisé);

— la commodité de réception (par exemple, les codes machines sont commodes pour le travail du processeur);

— la capacité maximale du canal de transmission;

— l'immunité contre les parasites;

— la réalisation de certaines propriétés de l'algorithme de codage (par exemple, simplicité du codage, décodage univoque), etc.

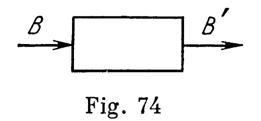


Fig. 74

Le *canal de transmission* peut être figuré par un organe à une entrée et une sortie (voir fig. 74). Le code du message B est appliqué à l'entrée. A la sortie on obtient le code du message B', où B' est un mot formé à l'aide d'un alphabet $\mathfrak{B}'$ et

$$B' = f(B).$$

Dans le cas le plus simple (un canal idéal sans bruit), c'est-à-dire dans le cas d'une ligne de transmission parfaite, $B' \equiv B$ (ou $f(B) = B$), donc $\mathfrak{B}' = \mathfrak{B}$. Dans le cas général, le canal de transmission peut assurer la traduction des codes, et $\mathfrak{B}' \neq \mathfrak{B}$ (comme dans un calculateur).

La *source de bruits* introduit des erreurs dans le canal de transmission d'où une déformation des codes à la sortie. Pour décrire la source de bruits on se sert de:

a) la *description logico-combinatoire* qui consiste à indiquer les restrictions assujetissant le nombre d'erreurs unitaires;

b) la *description statistique* qui se traduit par la donnée des caractéristiques probabilistes de la source.

Le *code du message à la sortie* est, dans le cas d'un canal sans bruit, un ensemble de mots formés à l'aide de l'alphabet $\mathfrak{B}' = \mathfrak{B}$.

Cependant la source de bruits peut conduire à une situation telle que

$$B' \neq B.$$

Le *message à la sortie* est un mot formé à l'aide d'un alphabet $\mathfrak{C}$. Dans le cas d'un canal sans bruit, c'est-à-dire lors d'une transmission idéale, $\mathfrak{C} = \mathfrak{A}$.

Pour passer des codes des messages à la sortie à ces messages mêmes à la sortie deux conversions sont nécessaires.

Correction du code du message à la sortie. Cette opération n'est possible que pour des codes de transmission spéciaux. Dans le cas où l'on a affaire à une transmission de messages, le passage a lieu de B' vers B.

Décodage. Il consiste à traduire le code obtenu par correction du code du message à la sortie, en le message à la sortie. Le décodage ne peut également être effectué que pour des codes de transmission spéciaux. Dans le cas d'une transmission des messages, le décodage est possible si existe l'application inverse F^{-1}.

Avec ce chapitre nous allons nous initier à la théorie du codage. Le but visé est de donner une idée sur:

a) les principales classes de codes;

b) les approches probabilistes et logico-combinatoires de description des problèmes;

c) la nature des problèmes à résoudre;

d) les méthodes de résolution de ces problèmes.

Dans les §§ 32 à 35, on étudie le codage alphabétique. L'exposé s'articule sur deux problèmes: la possibilité d'un décodage univoque et la construction de codes à redondance minimale. Dans le § 36 on aborde une classe de codes dits uniformes. On se penche sur le problème de l'élaboration de codes stables aux perturbations.

§ 32. Critère d'univocité du décodage

On étudie un codage alphabétique pour les alphabets $\mathfrak{A}$ et $\mathfrak{B}$ défini par le schéma:

$$\begin{matrix} a_1 - B_1, \\ \cdots \\ a_r - B_r, \end{matrix} \qquad (\Sigma)$$

et l'on admet que $S'(\mathfrak{A}) = S(\mathfrak{A})$, c'est-à-dire que la source débite l'ensemble de tous les mots formés à l'aide de l'alphabet $\mathfrak{A}$. Il est évident que le codage alphabétique engendre une application de l'ensemble $S(\mathfrak{A})$ dans l'ensemble $S(\mathfrak{B})$. Désignons par $S_\Sigma(\mathfrak{B})$ l'image de $S(\mathfrak{A})$ par cette application.

Le décodage est possible si l'application de $S(\mathfrak{A})$ sur $S_\Sigma(\mathfrak{B})$ est une bijection, autrement dit d'après le code B on peut restituer

de façon unique le message initial A dont le code est B (on dit encore que le code B est déchiffrable).

Exemple 2. Considérons un codage alphabétique pour lequel $\mathfrak{A} = \{a_1, a_2\}$, $\mathfrak{B} = \{b_1, b_2\}$ et le schéma Σ est de la forme

$$a_1 - b_1,$$
$$a_2 - b_1 b_2.$$

Supposons que B' et B'' sont les codes respectifs des mots A' et A''. Il est manifeste que si $A' \neq A''$, alors $B' \neq B''$.

Le décodage se déroule comme suit. Divisons le mot $B \in S_\Sigma(\mathfrak{B})$ en codes élémentaires. Pour cela on remarquera que chaque lettre b_2 du mot B est précédée de la lettre b_1. Ceci permet de regrouper tous les couples (b_1b_2). Le reste du mot B ne sera constitué que de lettres b_1. Si maintenant l'on remplace chaque couple (b_1b_2) par a_2 et chacune des lettres b_1 restantes par a_1, on obtient un mot A qui est l'image réciproque de B.

Supposons que $B = b_1b_1b_2b_1b_2b_1b_1b_1b_2$. En groupant les couples (b_1b_2) on décompose B en codes élémentaires:

$$B = b_1\,(b_1b_2)\,(b_1b_2)\,b_1b_1\,(b_1b_2),$$

d'où l'on déduit le mot

$$A = a_1a_2a_2a_1a_1a_2.$$

On pourrait exhiber une multitude d'exemples dans lesquels le codage alphabétique n'est pas biunivoque. La question qui se pose est de savoir s'il est possible de dire sur le vu du schéma Σ d'un codage alphabétique que ce codage est biunivoque ou non. La difficulté de ce problème tient au fait que la vérification immédiate de la biunivocité implique l'examen d'une infinité de mots.

Avant d'énoncer un critère général de biunivocité du codage alphabétique, considérons une condition suffisante relativement simple de biunivocité.

Définition. Supposons qu'un mot B est de la forme

$$B = B'B''.$$

Le mot B' s'appelle *début* ou *préfixe* de B, B'', *fin*.

Définition. On dit qu'un schéma Σ *définit un code sans préfixe* si pour tous i et j ($1 \leqslant i, j \leqslant r$, $i \neq j$) le mot B_i n'est pas un préfixe du mot B_j.

Théorème 1. *Si un schéma Σ définit un code sans préfixe, alors le codage alphabétique est biunivoque.*

Démonstration. Supposons qu'un mot B de $S_\Sigma(\mathfrak{B})$ se déchiffre de deux manières, donc admet deux partitions en codes élémentaires

$$B = B_{i_1} \dots B_{i_s},$$
$$B = B_{j_1} \dots B_{j_t}.$$

Supposons que $B_{i_1} = B_{j_1}, \dots, B_{i_{n-1}} = B_{j_{n-1}}, B_{i_n} \neq B_{j_n}$. Dans se cas l'un des mots B_{i_n}, B_{j_n} est le préfixe de l'autre. C. q. f. d.

L'exemple précédent montre que la condition d'être préfixe n'est pas nécessaire: Σ peut ne pas définir un code sans préfixe, et le code alphabétique défini par Σ être déchiffrable.

Supposons que $B = b_{i_1} \dots b_{i_n}$ est un mot de $S(\mathfrak{B})$. Désignons par $\tilde{B}$ le mot obtenu à partir de B par «inversion», c'est-à-dire que

$$\tilde{B} = b_{i_n} \dots b_{i_1}.$$

Désignons par $\tilde{\Sigma}$ le schéma

$$a_1 - \tilde{B}_1,$$
$$\dots\dots$$
$$a_r - \tilde{B}_r.$$

Exemple 3. Prenons pour Σ le schéma de l'exemple 2. Alors $\tilde{\Sigma}$ est de la forme

$$a_1 - b_1,$$
$$a_2 - b_2 b_1.$$

$\tilde{\Sigma}$ définit un code sans préfixe, et, en vertu du théorème 1, le codage alphabétique sera biunivoque.

Remarque. Les codages alphabétiques définis par les schémas Σ et $\tilde{\Sigma}$ sont ou ne sont pas simultanément biunivoques.

Cette remarque permet de renforcer le théorème 1.

Théorème 2. *Si l'un des schémas Σ et $\tilde{\Sigma}$ définit un code sans préfixe, alors le codage alphabétique défini par Σ (resp. $\tilde{\Sigma}$) sera biunivoque.*

On peut exhiber un exemple de codage alphabétique de schéma Σ tel que Σ et $\tilde{\Sigma}$ ne définissent pas un code sans préfixe et le codage alphabétique soit biunivoque.

L'exemple précédent ne convient plus pour cela, mais il peut être légèrement perfectionné.

Exemple 4. Soient $\mathfrak{A} = \{a_1, a_2, a_3\}$ et $\mathfrak{B} = \{b_1, b_2, b_3\}$. Considérons le schéma Σ :

$$\begin{aligned} &a_1 - b_1, \\ &a_2 - b_1 b_2, \\ &a_3 - b_3 b_1. \end{aligned} \qquad (\Sigma)$$

Il est évident que Σ et $\tilde{\Sigma}$ ne définissent pas de code sans préfixe, bien que le codage alphabétique soit biunivoque. En effet si $B \in S_\Sigma(\mathfrak{B})$, ce mot se décompose de façon unique en codes élémentaires:

la lettre b_2 est immédiatement précédée de la lettre b_1: distinguons le couple $(b_1 b_2)$;

la lettre b_3 est immédiatement suivie de la lettre b_1: distinguons le couple $(b_3 b_1)$;

une fois isolés tous les couples $(b_1 b_2)$ et $(b_3 b_1)$, il ne restera plus que les symboles b_1.

On supposera dans la suite que les codes élémentaires de Σ sont deux à deux distincts.

Avant de poursuivre notre exposé, introduisons quelques notations: désignons par $l(B)$ la longueur du mot B, c'est-à-dire le

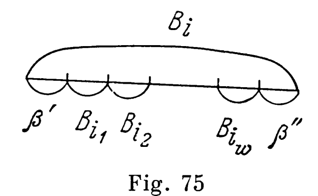


Fig. 75

nombre de lettres de ce mot. En particulier, on admettra que $l(B_i) = l_i$ pour les codes élémentaires B_i $(i = 1, \ldots, r)$. Appelons L la quantité $l(B_1 \ldots B_r)$, c'est-à-dire la « longueur » du schéma Σ.

Soit

$$B_i = \beta' B_{i_1} \ldots B_{i_w} \beta'' \qquad (1)$$

une décomposition non triviale du code élémentaire B_i, c'est-à-dire une décomposition différente de la décomposition

$$B_i = B_i \quad (\beta' = \beta'' = \Lambda).$$

Dans cette décomposition on admet que

a) β' ne peut avoir un code élémentaire pour fin.

b) β'' ne peut avoir un code élémentaire pour préfixe.

Le paramètre w peut être un nombre entier, positif ou nul.

La relation (1) signifie que dans le code élémentaire B_i on peut supprimer un préfixe β' et une fin β'' de telle sorte que la partie restante se décompose en codes élémentaires (voir fig. 75).

Il est évident que tout B_i admet un nombre fini de décompositions de la forme (1). Désignons par W le maximum des nombres w, pris sur toutes les décompositions de B_i et sur tous les i, c'est-à-dire que

$$W = \max w.$$

Exemple 5. Considérons le codage alphabétique défini par $\mathfrak{A} = \{a_1, a_2, a_3, a_4, a_5\}$, $\mathfrak{B} = \{b_1, b_2, b_3\}$ et le schéma

$$\begin{aligned} &a_1 - b_1b_2,\\ &a_2 - b_1b_3b_2,\\ &a_3 - b_2b_3,\\ &a_4 - b_1b_2b_1b_3,\\ &a_5 - b_2b_1b_2b_2b_3. \end{aligned}$$

Comme $6 > l_i \geqslant 2$, on a $W < 3$. D'autre part

$$B_5 = b_2b_1b_2b_2b_3 = b_2B_1B_3,$$

donc $W = 2$.

Désignons enfin par $S^N(\mathfrak{A})$ l'ensemble de tous les mots de longueur $\leqslant N$, formés à l'aide de l'alphabet $\mathfrak{A}$. Il est clair que $S^N(\mathfrak{A})$ est un ensemble fini dont le cardinal est $\sum_{i=1}^{N} r^i$.

Formulons maintenant un critère de biunivocité du codage alphabétique.

Théorème 3 (Markov Al. [12, 13]). *Pour tout codage alphabétique défini par un schéma Σ il existe un N_0 tel que la biunivocité du codage alphabétique se ramène à celle du codage d'un ensemble fini $S^{N_0}(\mathfrak{A})$ et*

$$N_0 \leqslant \left[\frac{(W+1)(L-r+2)}{2}\right].$$

Démonstration. Si la biunivocité d'un codage alphabétique est violée, il existe un nombre B qui admet au moins deux décodages différents A' et A''. Pour prouver le théorème il suffit de montrer qu'on peut trouver un nombre B tel que A' et A'' soient justiciables des inégalités

$$l(A'),\ l(A'') \leqslant \left[\frac{(W+1)(L-r+2)}{2}\right].$$

Un mot B admettant au moins deux décodages est dit *irréductible* si chaque mot B' déduit à partir de B par suppression d'une tranche non vide admet un décodage au plus.

Il est clair que, dès le départ, on peut supposer que le mot B est irréductible. Considérons ses deux décodages A' et A''. Il est évident qu'ils donnent lieu à deux décompositions du mot B en codes élé-

mentaires: la décomposition supérieure T_1 et la décomposition inférieure T_2 (voir fig. 76).

Considérons le produit T de ces décompositions, c'est-à-dire la décomposition obtenue par la décomposition simultanée de T_1 et de T_2. Répartissons les mots de la partition T en deux classes: dans la première on portera les mots qui sont des codes élémentaires, dans la seconde tous les autres. Il est évident que chaque mot de la première classe figure exactement dans une partition, tandis que les mots de la deuxième classe (ils sont représentés par des traits gras sur la

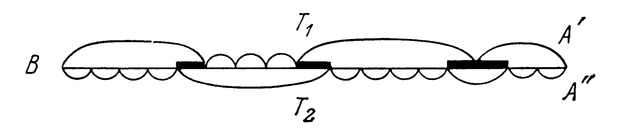


Fig. 76

figure 76) sont à la fois les préfixes et les fins non vides de codes élémentaires appartenant à des partitions différentes. Montrons que les mots de la deuxième classe sont deux à deux distincts. Supposons le contraire, c'est-à-dire que $\beta' = \beta'' = \beta$. Alors

$$B = B'\beta'B''\beta''B''' = B'\beta B''\beta B'''.$$

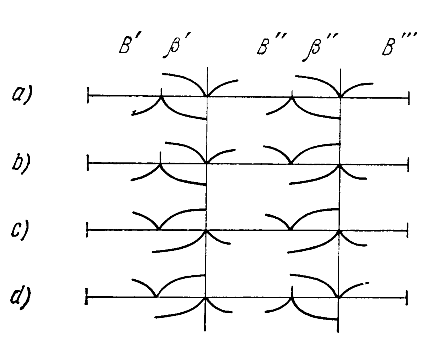


Fig. 77

On affirme que le mot $B'\beta'B'''$ qui se déduit à partir de B par suppression de la tranche $B''\beta''$, admet au moins deux décodages. Les mots β' et β'' peuvent être disposés de quatre manières qui sont représentées sur la figure 77. Il est immédiat de voir que c) se ramène à a), et d) à b). Pour cela il suffit dans c) et d) de modifier les rôles de T_1 et T_2.

Dans a), en éliminant la tranche $B''\beta''$ et en rapprochant les tranches $B'\beta'$ et B''' on obtient la partition supérieure par juxtaposition des tranches de T_1, et la partition inférieure, par juxtaposition des tranches de T_2.

Dans b), la partition supérieure s'obtient par juxtaposition d'une tranche de la partition supérieure de $B'\beta'$ et de la partition inférieure de B''', la partition inférieure, par juxtaposition de la partition inférieure de $B'\beta'$ et de la partition supérieure de B'''.

Donc, dans tous les cas on obtient au moins deux décodages pour le mot $B'\beta'B'''$, ce qui contredit l'irréductibilité du mot initial.

Le nombre p de mots de la deuxième classe n'est pas supérieur à celui des préfixes non vides des codes élémentaires, c'est-à-dire que

$$p < (l(B_1) - 1) + (l(B_2) - 1) + \ldots + (l(B_r) - 1) = L - r.$$

Les mots de la deuxième classe partagent le mot B en $L - r + 1$ tranches au plus dont certaines sont éventuellement vides (voir fig. 78).

Soit $\beta^0 = \beta^{p+1} = \Lambda$. Considérons l'une des tranches comprises entre les mots β^j et β^{j+1} $(j = 0, \ldots, p)$:

$$\beta^j B_{i_1} \ldots B_{i_w} \beta^{j+1}.$$

Tous les mots $B_{i_1}, \ldots, B_{i_w}$ de cette tranche appartiennent à une même partition du mot B, par exemple à T_1, et le mot

$$\beta^j B_{i_1} \ldots B_{i_w} \beta^{j+1} = B_i$$

à une autre partition, par exemple à T_2. Ceci étant, $w \leqslant W$. Donc,

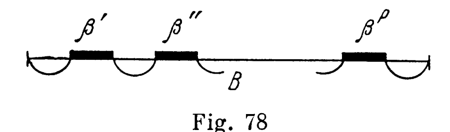


Fig. 78

à chaque tranche sont liés w codes élémentaires d'une partition et un code élémentaire de l'autre partition.

Si maintenant l'on prend deux tranches adjacentes

$$\beta^j B_{i'_1} \ldots B_{i'_{w'}} \beta^{j+1} B_{i''_1} \ldots B_{i''_{w''}} \beta^{j+2},$$

alors les codes élémentaires

$$B_{i'_1}, \ldots, B_{i'_{w'}}, \quad B_{i''} = \beta^{j+1} B_{i''_1} \ldots B_{i''_{w''}} \beta^{j+2}$$

appartiennent à une partition et les codes

$$B_{i''_1}, \ldots, B_{i''_{w''}}, \quad B_{i'} = \beta^j B_{i'_1} \ldots B_{i'_{w'}} \beta^{j+1}$$

à l'autre (voir fig. 79).

Par suite, aux tranches de la partition T comprises entre les mots β^j et β^{j+1} (la parité de j étant la même) correspondent les codes élémentaires B_i de la partition initiale (par exemple T_1) et les groupes des codes $B_{i_1}, \ldots, B_{i_w}$ de la même partition pour une autre parité de j. De là on déduit sans peine une majoration pour le nombre maximal de codes élémentaires appartenant à la partition. Ce nombre n'est pas supérieur à

$$\left[\frac{L-r+2}{2}\right] \cdot 1 + \left[\frac{L-r+2}{2}\right] W \leqslant \left[\frac{(W+1)(L-r+2)}{2}\right].$$

On obient

$$l(A'),\ l(A'') \leqslant \left[\frac{(W+1)(L-r+2)}{2}\right].$$

Si maintenant l'on pose $N_0 = \max(l(A'), l(A''))$, il est évident que la biunivocité du codage est déjà violée sur l'ensemble $S^{N_0}(\mathfrak{A})$, puisque $A', A'' \in S^{N_0}(\mathfrak{A})$. Ce qui prouve le théorème.

Le critère d'univocité du décodage nous fournit un algorithme simple de reconnaissance de la biunivocité du codage alphabétique

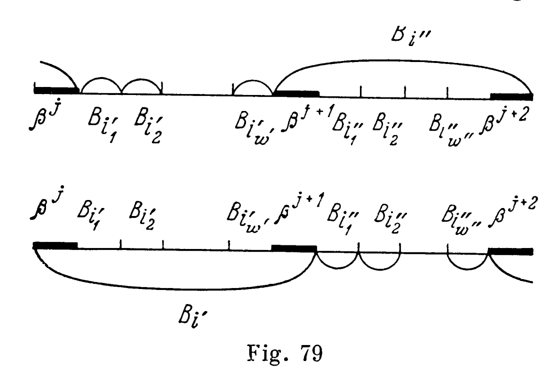


Fig. 79

sur le vu du schéma Σ. Il suffit pour cela de considérer l'ensemble des mots de longueur $\leqslant N_0$, formés à l'aide de l'alphabet $\mathfrak{A}$, c'est-à-dire l'ensemble $S^{N_0}(\mathfrak{A})$, et de voir si le codage de cet ensemble fini est biunivoque. La taille de cet algorithme est *grosso modo* de l'ordre de r^{N_0}. Il s'avère que cet algorithme ne peut pas être utilisé même dans les exemples simples.

Exemple 6. Considérons un codage alphabétique (voir exemple 5). On a $r=5$, $W=2$, $L=16$. Si l'on prend $N_0 = \left[\frac{(W+1)(L-r+2)}{2}\right] = \left[\frac{3\cdot 13}{2}\right] = 19$, on obtient

$$r^{N_0} = 5^{19},$$

c'est-à-dire un nombre très grand.

§ 33. Algorithme de reconnaissance de l'univocité du décodage

De la démonstration du critère d'univocité du décodage on peut extraire un algorithme assez efficace qui permet de dire si le décodage est possible. Cet algorithme se formule en termes de théorie des graphes (Markov Al. [12, 13, 14]).

Supposons qu'un codage alphabétique est défini par le schéma Σ :

$$\begin{array}{c} a_1 - B_1, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ a_r - B_r. \end{array} \qquad (\Sigma)$$

Pour chaque code élémentaire B_i considérons toutes les représentations non triviales de la forme

$$B_i = \beta' B_{i_1} \ldots B_{i_w} \beta''. \qquad (2)$$

Désignons par $\mathfrak{B}_0$ l'ensemble contenant

a) le mot vide Λ;

b) les mots β que l'on rencontre dans les décompositions de la forme (2) aussi bien sous la forme de préfixes que de fins.

A chaque mot de $\mathfrak{B}_0$ associons un point du plan.

Soient β', $\beta'' \in \mathfrak{B}_0$. Considérons toutes les décompositions de la forme

$$B_i = \beta' B_{i_1} \ldots B_{i_w} \beta''.$$

Relions les sommets correspondant aux mots β' et β'' par un segment orienté (de β' vers β'') auquel on affecte le mot $B_{i_1} \ldots B_{i_w}$. Désignons par $\Gamma(\Sigma)$ le graphe obtenu.

Théorème 4. *Une condition nécessaire et suffisante pour que le codage alphabétique défini par le schéma Σ ne soit pas biunivoque est que $\Gamma(\Sigma)$ contienne un circuit passant par le sommet Λ.*

Démonstration. *Condition nécessaire.* Supposons que le codage alphabétique n'est pas biunivoque. Il existe alors un mot irréductible B de la forme (voir § 32)

$$B = B_{i_1^0} \ldots B_{i_{w0}^0} \beta' B_{i_1^1} \ldots B_{i_{w1}^1} \beta'' \ldots \beta^p B_{i_1^p} \ldots B_{i_{wp}^p},$$

tel que

$$\begin{array}{l} B_{i^0} = B_{i_1^0} \ldots B_{i_{w0}^0} \beta^1, \\ B_{i^1} = \beta' B_{i_1^1} \ldots B_{i_{w1}^1} \beta'', \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ B_{i^p} = \beta^p B_{i_2^p} \ldots B_{i_{wp}^p}. \end{array}$$

Donc, le graphe $\Gamma(\Sigma)$ contient un circuit (voir fig. 80) passant par le sommet Λ.

Condition suffisante. Supposons que $\Gamma(\Sigma)$ contienne un circuit passant par le sommet Λ (voir fig. 81). Alors le mot B, où

$$B = B_{i_1^0} \ldots B_{i_{w0}} \beta' B_{i_1^1} \ldots B_{i_{w1}^1} \beta'' \ldots \beta^p B_{i_1^p} \ldots B_{i_{wp}^p}$$

admet deux décodages définis par les partitions

$$B=(B_{i_1^0})\dots(B_{i_{w0}^0})(\beta' B_{i_1^1}\dots B_{i_{w1}^1}\beta'')(B_{i_1^2})\dots(B_{i_{w2}^2})(\beta''' B_{i_1^3}\dots B_{i_{w3}^3}\beta^{IV})\dots,$$

$$B=(B_{i_1^0}\dots B_{i_{w0}^0}\beta')(B_{i_1^1})\dots(B_{i_{w1}^1})(\beta'' B_{i_1^2}\dots B_{i_{w2}^2}\beta''')\dots$$

Ce qui prouve le théorème.

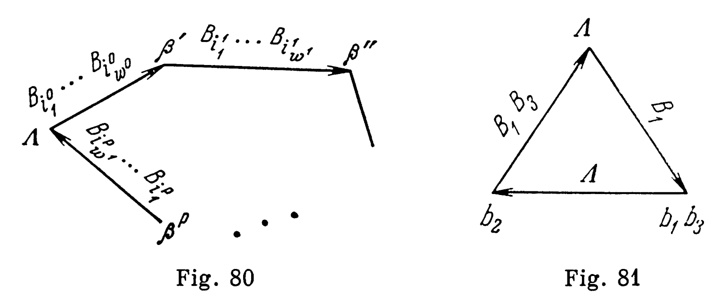


Fig. 80 Fig. 81

Donc, l'algorithme consiste à construire le graphe $\Gamma(\Sigma)$ et à trouver les circuits passant par le sommet Λ.

Exemple 7. Considérons le codage alphabétique (cf. exemple 5) défini par le schéma

$$\begin{aligned} a_1 &- b_1b_2,\\ a_2 &- b_1b_3b_2,\\ a_3 &- b_2b_3,\\ a_4 &- b_1b_2b_1b_3,\\ a_5 &- b_2b_1b_2b_2b_3. \end{aligned} \qquad (\Sigma)$$

On a les décompositions non triviales:

$$B_1=(b_1)(b_2),$$
$$B_2=(b_1)(b_3b_2)=(b_1b_3)\,b_2,$$
$$B_3=(b_2)(b_3),$$
$$B_4=(b_1)(b_2b_1b_3)=(b_1b_2)(b_1b_3)=(b_1b_2b_1)(b_3),$$
$$B_5=(b_2)(b_1b_2b_2b_3)=(b_2)(b_1b_2)(b_2b_3)=(b_2b_1)(b_2b_2b_3)=$$
$$=(b_2b_1b_2)(b_2b_3)=(b_2b_1b_2b_2)(b_3).$$

Il est évident que $\mathfrak{B}_0=\{\Lambda,\ b_2,\ b_1b_3\}$ et qu'à cet ensemble sont liées les décompositions

$$B_2=(b_1b_3)(b_2),$$
$$B_4=(b_1b_2)(b_1b_3)=B_1(b_1b_3),$$
$$B_5=(b_2)(b_1b_2)(b_2b_3)=(b_2)B_1B_3.$$

Ceci permet de construire le graphe $\Gamma(\Sigma)$ (voir fig. 81). Ce graphe contient un circuit engendrant le mot B

$$B = B_1 b_1 b_3 b_2 B_1 B_3,$$

qui admet deux décodages:

$$B = (B_1 b_1 b_3)(b_2 B_1 B_3), \quad \text{i.e.} \quad A' = a_4 a_5,$$

$$B = B_1 (b_1 b_3 b_2) B_1 B_3, \quad \text{i.e.} \quad A'' = a_1 a_2 a_1 a_3.$$

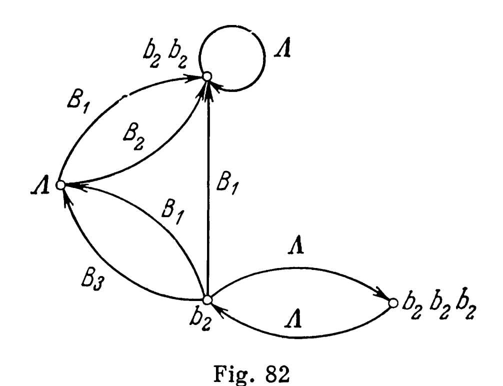


Fig. 82

Exemple 8. Considérons le codage alphabétique défini par le schéma

$$\begin{aligned} a_1 &- b_1, \\ a_2 &- b_2 b_1, \\ a_3 &- b_1 b_2 b_2, \\ a_4 &- b_2 b_1 b_2 b_2, \\ a_5 &- b_2 b_2 b_2 b_2. \end{aligned} \qquad (\Sigma)$$

On a les décompositions non triviales suivantes:

$B_2 = (b_2)\, b_1 = b_2 B_1$;

$B_3 = (b_1)(b_2 b_2) = B_1 (b_2 b_2)$; $B_3 = (b_1 b_2)(b_2)$;

$B_4 = (b_2)(b_1)(b_2 b_2) = (b_2) B_1 (b_2 b_2)$; $B_4 = (b_2)(b_1 b_2 b_2) = b_2 B_3$;

$B_4 = (b_2 b_1)(b_2 b_2) = B_2 (b_2 b_2)$; $B_4 = (b_2 b_1 b_2)(b_2)$;

$B_5 = (b_2)(b_2 b_2 b_2) = (b_2 b_2)(b_2 b_2) = (b_2 b_2 b_2)(b_2)$.

Il est évident que $\mathfrak{B}_0 = \{\Lambda, b_2, b_2b_2, b_2b_2b_2\}$ et qu'à cet ensemble sont liées les décompositions

$$B_2 = b_2B_1;$$
$$B_3 = B_1(b_2b_2);$$
$$B_4 = (b_2)B_1(b_2b_2); \quad B_4 = b_2B_3; \quad B_4 = B_2(b_2b_2);$$
$$B_5 = (b_2)(b_2b_2b_2) = (b_2b_2)(b_2b_2) = (b_2b_2b_2)(b_2).$$

On obtient un graphe $\Gamma(\Sigma)$ (voir fig. 82) ne contenant pas de circuit passant par le sommet Λ. Donc, le codage alphabétique défini par le schéma Σ est biunivoque. Cette conclusion ne peut être tirée du théorème 2.

§ 34. Sur une propriété des codes déchiffrables

Les codes déchiffrables tiennent une place très importante en codage alphabétique. Nous allons démontrer deux propositions pour de tels codes.

Soit le codage alphabétique défini par le schéma Σ

$$\begin{matrix} a_1 - B_1, \\ \cdot\ \cdot\ \cdot\ \cdot \\ a_r - B_r. \end{matrix} \qquad (\Sigma)$$

Désignons par q la valence de l'alphabet $\mathfrak{B}$ et $l_i = l(B_i)$ $(i = 1, \ldots, r)$.

Théorème 5 (inégalité de McMillan [16]). *Si le codage alphabétique défini par le schéma Σ est biunivoque, alors*

$$\sum_{i=1}^{r} \frac{1}{q^{l_i}} \leqslant 1.$$

Démonstration. Considérons tous les mots de longueur n formés à l'aide de l'alphabet $\mathfrak{A}$. Ils peuvent tous être engendrés par l'expression

$$(a_1 + \ldots + a_r)^n,$$

si l'on multiplie les parenthèses (par exemple à gauche) sans se servir de la commutativité et que l'on considère le produit

$$a_{i_1}a_{i_2} \ldots a_{i_n}$$

comme un mot de $S(\mathfrak{A})$. Il est évident que le symbole a_{i_1} appartiendra à la première parenthèse, a_{i_2}, à la seconde, etc., a_{i_n}, à la n-ième. On a

$$(a_1 + \ldots + a_r)^n = \sum_{(i_1 i_2 \ldots i_n)} a_{i_1}a_{i_2} \ldots a_{i_n}.$$

On obtient les codes correspondant à ces mots si l'on remplace les symboles $a_1, \ldots, a_r$ par les codes élémentaires $B_1, \ldots, B_r$ en se servant du schéma du codage alphabétique. On obtient

$$(B_1 + \ldots + B_r)^n = \sum_{(i_1 i_2 \ldots i_n)} B_{i_1} B_{i_2} \ldots B_{i_n}. \tag{3}$$

Le codage alphabétique étant biunivoque, si $(i_1, \ldots, i_n) \neq \neq (j_1, \ldots, j_n)$, c'est-à-dire si $a_{i_1} \ldots a_{i_n} \neq a_{j_1} \ldots a_{j_n}$, alors

$$B_{i_1} \ldots B_{i_n} \neq B_{j_1} \ldots B_{j_n}.$$

A l'identité (3) est associée l'identité

$$\left(\frac{1}{q^{l_1}} + \ldots + \frac{1}{q^{l_r}}\right)^n = \sum_{(i_1 \ldots i_n)} \frac{1}{q^{l_{i_1} + \ldots + l_{i_n}}}. \tag{4}$$

Il est évident qu'aux termes ayant même dénominateur de la somme de droite correspondent dans (3) les mots $B_{i_1} B_{i_2} \ldots B_{i_n}$ de même longueur. Introduisons les notations:

$$t = l_{i_1} + \ldots + l_{i_n},$$

$\nu(n, t)$, nombre des mots $B_{i_1} B_{i_2} \ldots B_{i_n}$ de (3) de longueur t *) et

$$l = \max_{1 \leqslant i \leqslant r} l_i.$$

On obtient

$$\sum_{(i_1 \ldots i_n)} \frac{1}{q^{l_{i_1} + \ldots + l_{i_n}}} = \sum_{t=1}^{nl} \frac{\nu(n, t)}{q^t}.$$

La biunivocité du codage alphabétique entraîne que

$$\nu(n, t) \leqslant q^t,$$

donc

$$\sum_{t=1}^{nl} \frac{\nu(n, t)}{q^t} \leqslant nl.$$

En combinant la dernière inégalité avec (4), on trouve

$$\sum_{i=1}^{r} \frac{1}{q^{l_i}} \leqslant \sqrt[n]{nl}.$$

Cette inégalité est valable pour tout n. Comme son premier membre ne dépend pas de n, elle est valable pour $n \to \infty$.

On obtient en définitive

$$\sum_{i=1}^{r} \frac{1}{q^{l_i}} \leqslant 1.$$

---

*) $\nu(n, t) = 0$ s'il n'existe pas de mot de longueur t dans (3).

Ce qui achève la démonstration du théorème. Cette proposition est complétée par le

Théorème 6. *Si le codage alphabétique défini par le schéma*

$$\begin{matrix} a_1 - B_1, \\ \cdot\ \cdot\ \cdot\ \cdot \\ a_r - B_r \end{matrix} \qquad (\Sigma)$$

est biunivoque, il existe alors un codage alphabétique donné par le schéma

$$\begin{matrix} a_1 - B'_1, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ a_r - B'_r, \end{matrix} \qquad (\Sigma')$$

qui définit un code sans préfixe. De plus, pour les codes élémentaires $B'_1, \ldots, B'_r$ *qui sont des mots de* $S(\mathfrak{B})$ *on a les relations*

$$l(B'_1) = l(B_1), \ \ldots, \ l(B'_r) = l(B_r).$$

Démonstration. On peut admettre que $l_1 \leqslant \ldots \leqslant l_r$. D'après la biunivocité du codage alphabétique et le théorème 5 on a

$$\sum_{i=1}^{r} \frac{1}{q^{l_i}} \leqslant 1.$$

Répartissons les nombres $l_1, \ldots, l_r$ en classes de telle sorte que l_i et l_j appartiennent à une même classe si et seulement si $l_i = l_j$. Supposons que μ $(1 \leqslant \mu \leqslant r)$ désigne le nombre de classes, et $\lambda_1, \ldots, \lambda_\mu, \nu_1, \ldots, \nu_\mu$, respectivement les représentants et les cardinaux de ces classes. On peut admettre de même que

$$\lambda_1 < \lambda_2 < \ldots < \lambda_\mu.$$

L'inégalité de McMillan s'écrit encore

$$\sum_{i=1}^{\mu} \frac{\nu_i}{q^{\lambda_i}} \leqslant 1.$$

Cette inégalité engendre de nombreuses inégalités auxiliaires

$$\frac{\nu_1}{q^{\lambda_1}} \leqslant 1 \quad \text{ou} \quad \nu_1 \leqslant q^{\lambda_1},$$

$$\frac{\nu_1}{q^{\lambda_1}} + \frac{\nu_2}{q^{\lambda_2}} \leqslant 1 \quad \text{ou} \quad \nu_2 \leqslant q^{\lambda_2} - \nu_1 q^{\lambda_2 - \lambda_1},$$

$$\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot$$

$$\frac{\nu_1}{q^{\lambda_1}} + \frac{\nu_2}{q^{\lambda_2}} + \ldots + \frac{\nu_\mu}{q^{\lambda_\mu}} \leqslant 1 \quad \text{ou}$$

$$\nu_\mu \leqslant q^{\lambda_\mu} - \nu_1 q^{\lambda_\mu - \lambda_1} - \nu_2 q^{\lambda_\mu - \lambda_2} - \ldots - \nu_{\mu-1} q^{\lambda_\mu - \lambda_{\mu-1}}.$$

Considérons les mots de longueur λ_1 de $S(\mathfrak{B})$. Comme $\nu_1 \leqslant q^{\lambda_1}$, on peut choisir parmi eux ν_1 mots de longueur λ_1. Désignons-les par $B'_1, \ldots, B'_{\nu_1}$. Supprimons les mots qui commencent par $B'_1, \ldots, B'_{\nu_1}$. Considérons ensuite l'ensemble des mots de longueur λ_2 qui ne débutent pas par $B'_1, \ldots, B'_{\nu_1}$. Il est évident que ces mots sont au nombre de $q^{\lambda_2} - \nu_1 q^{\lambda_2 - \lambda_1}$. Comme $\nu_2 \leqslant q^{\lambda_2} - \nu_1 q^{\lambda_2 - \lambda_1}$, on peut extraire de cet ensemble ν_2 mots, que nous désignerons par $B'_{\nu_1+1}, \ldots, B'_{\nu_1+\nu_2}$. Excluons les mots commençant par $B'_{\nu_1+1}, \ldots, B'_{\nu_1+\nu_2}$ et ainsi de suite. En utilisant successivement les inégalités auxiliaires, on construit les mots $B'_1, \ldots$ $\ldots, B'_r \left(r = \sum_{i=1}^{\mu} \nu_i\right)$. Si on les prend pour codes élémentaires, on obtient le schéma cherché Σ'. Ce schéma définit un code sans préfixe et

$$l(B'_1) = l(B_1), \ \ldots, \ l(B'_r) = l(B_r).$$

Ce qui achève la démonstration du théorème.

§ 35. Codes à redondance minimale

Soient donnés un alphabet $\mathfrak{A} = \{a_1, \ldots, a_r\}$ $(r \geqslant 2)$ et les probabilités $p_1, \ldots, p_r$ $\left(\sum_{i=1}^{r} p_i = 1\right)$ d'apparitions des symboles $a_1, \ldots, a_r$. Soit donné par ailleurs un alphabet $\mathfrak{B} = \{b_1, \ldots, b_q\}$ $(q \geqslant 2)$. On peut alors construire plusieurs schémas

$$\begin{array}{c} a_1 - B_1, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ a_r - B_r, \end{array} \qquad (\Sigma)$$

définissant des codes déchiffrables. En particulier, on peut toujours prendre pour codes $B_1, \ldots, B_r$ des mots de longueur l de $S(\mathfrak{B})$, où $l = \,]\log_q r[$.

Pour chaque schéma Σ on peut introduire une quantité l_{m} appelée *redondance du codage*, qui est définie comme l'espérance mathématique de la longueur d'un code élémentaire, c'est-à-dire que

$$l_{\mathrm{m}} = \sum_{i=1}^{r} p_i l_i, \quad l_i = l(B_i).$$

Il est évident que la quantité l_{m} $(l_{\mathrm{m}} \geqslant 1)$ indique de combien de fois s'accroît la longueur d'un mot par un codage défini par le schéma Σ.

Exemple 9. Soient $r = 4$, $q = 2$ et $p_1 = 0{,}40$, $p_2 = 0{,}25$, $p_3 = 0{,}20$, $p_4 = 0{,}15$.

Considérons deux schémas

$$\begin{matrix} a_1\text{—}00, \\ a_2\text{—}01, \\ a_3\text{—}10 \\ a_4\text{—}11, \end{matrix} \quad (\Sigma') \qquad \begin{matrix} a_1\text{—}1, \\ a_2\text{—}01, \\ a_3\text{—}000 \\ a_4\text{—}001. \end{matrix} \quad (\Sigma'')$$

Il est évident qu'ils définissent des codes déchiffrables. Trouvons leur redondance :

$$l'_{\mathrm{m}} = 2, \quad l''_{\mathrm{m}} = 0{,}40 + 2 \cdot 0{,}25 + 3 \cdot 0{,}35 = 1{,}95.$$

Donc, la redondance varie lorsqu'on passe d'un schéma à l'autre. Ceci nous conduit à introduire pour la source d'information la quantité l_*, où

$$l_* = \inf_{\Sigma} l_{\mathrm{m}}^{\Sigma}$$

(la borne inférieure est prise sur tous les schémas Σ qui assurent un codage biunivoque).

Il est immédiat de voir que

$$1 \leqslant l_* \leqslant]\log_q r[.$$

On voit que pour construire des codes dont la quantité l_{m} est voisine de l_* on peut ne pas tenir compte des codes dont l_{m} est plus grande que $]\log_q r[$. Donc, pour de tels schémas

$$p_i l_i \leqslant]\log_q r[.$$

Comme les termes dont $p = 0$ ne jouent aucun rôle dans le calcul de l_{m}, en posant $p_* = \min\limits_{\substack{i \\ p_i \neq 0}} p_i$, on obtient

$$l_i \leqslant \frac{]\log_q r[}{p_*}$$

pour tous les i tels que $p_i \neq 0$. Par suite, le nombre de variantes de valeurs de l_{m} pour lesquelles $l_* \leqslant l_{\mathrm{m}} \leqslant]\log_q r[$ est fini. Donc, la quantité l_* est réalisée sur un Σ et peut être définie comme

$$\min_{\Sigma} l_{\mathrm{m}}^{\Sigma}.$$

Définition. Les codes définis par le schéma Σ avec $l_{\mathrm{m}} = l_*$ s'appellent *codes à redondance minimale* ou codes de Huffman [1].

Il est évident que les codes à redondance minimale donnent lieu en moyenne à un accroissement minimal de la longueur des mots par codage. Il y a donc intérêt à étudier le problème de la construction de codes à redondance minimale.

Remarque. Le théorème de codage alphabétique biunivoque affirme l'existence d'un codage alphabétique définissant un code sans préfixe qui fournit des codes à redondance minimale. Donc, pour construire des codes à redondance minimale on peut se contenter de codes sans préfixe.

Passons maintenant à la construction de codes à redondance minimale. A chaque codage alphabétique définissant un code sans préfixe on peut associer un arbre de code. Illustrons ceci sur un exemple. Soit donné un schéma de la forme suivante ($r = 8$, $q = 4$):

$$
\begin{array}{ll}
a_1 - b_1b_3 & p_1 = 0{,}22 \\
a_2 - b_3 & p_2 = 0{,}20 \\
a_3 - b_1b_1 & p_3 = 0{,}14 \\
a_4 - b_1b_2 & p_4 = 0{,}11 \\
a_5 - b_4b_2b_3 & p_5 = 0{,}10 \\
a_6 - b_1b_4 & p_6 = 0{,}09 \\
a_7 - b_4b_1 & p_7 = 0{,}08 \\
a_8 - b_4b_2b_4 & p_8 = 0{,}06.
\end{array}
$$

Il est évident que ce schéma définit un code sans préfixe et que

$$
l_{\mathrm{m}} = 0{,}20 + 2\,(0{,}22 + 0{,}14 + 0{,}11 + 0{,}09 + 0{,}08) + \\
+ 3\,(0{,}10 + 0{,}06) = 0{,}20 + 2\cdot 0{,}64 + 3\cdot 0{,}16 = 0{,}20 + \\
+ 1{,}28 + 0{,}48 = 1{,}96.
$$

Les codes élémentaires définissent un arbre (voir fig. 83).

Aux sommets terminaux de cet arbre correspondent des codes élémentaires définis par un chemin (une branche) issu de la racine, et auxquels sont affectées les probabilités d'apparition des codes élémentaires. Il est immédiat de voir que l'arbre dont les sommets terminaux sont affectés de probabilités détermine un schéma de codage alphabétique définissant un code sans préfixe.

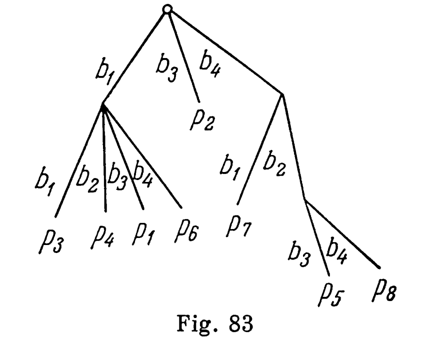


Fig. 83

Considérons d'abord un cas particulier, notamment le cas où l'un au plus des nombres $p_1, \ldots, p_r$ est nul. Sans nuire à la généralité on peut admettre que

$$p_1 \geqslant p_2 \geqslant \ldots \geqslant p_r.$$

Lemme 1. *Pour un code à redondance minimale, la condition* $p_j < p_i$ *entraîne que* $l_j \geqslant l_i$.

Démonstration. Supposons le contraire, c'est-à-dire que $p_j < p_i$ et $l_j < l_i$. Alors, si dans le schéma Σ :

$$\begin{matrix} \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ a_i - B_i, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ a_j - B_j, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \end{matrix} \qquad (\Sigma)$$

on permute les codes élémentaires B_i et B_j, on obtient le schéma Σ' :

$$\begin{matrix} \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ a_i - B_j, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ a_j - B_i, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \end{matrix} \qquad (\Sigma')$$

de redondance l'_{m} inférieure à celle du schéma Σ. En effet,

$$l_{\text{m}} - l'_{\text{m}} = (p_i l_i + p_j l_j) - (p_i l_j + p_j l_i) = (p_i - p_j)(l_i - l_j) > 0.$$

Ceci contredit la redondance minimale de Σ. Ce qui prouve le lemme.

Corollaire. *Dans un arbre pour un code à redondance minimale les probabilités affectées aux sommets terminaux du* l'*-ième étage ne sont pas inférieures à celles affectées aux sommets terminaux du* l''*-ième étage si* $l'' > l'$.

On étudiera dans la suite des arbres finis de degré maximal q (on appelle degré d'un sommet le nombre d'arêtes auxquelles appartient ce sommet).

Définition. On dit qu'un arbre fini est *saturé* si les degrés de tous ses sommets à l'exception éventuellement d'un seul situé à l'avant-dernier étage, sont égaux à 0 ou à q, et le degré du sommet excepté est égal à q_0, où $2 \leqslant q_0 < q$.

Il est immédiat de voir que le nombre q_0 se définit de façon unique à partir de l'équation

$$r = t(q-1) + q_0.$$

Calculons le reste de la division de r par $q - 1$ et convenons *) que

$$q_0 = \begin{cases} q-1 & \text{si le reste est égal à } 0 \\ \text{au reste} & \text{si celui-ci est} \geqslant 2. \end{cases} \qquad (5)$$

*) Le reste est $\neq 1$ s'il existe un sommet excepté.

Lemme 2. *Sous les contraintes* (5) *il existe un code à redondance minimale dont l'arbre est saturé.*

Démonstration. Considérons deux transformations des arbres du type ci-dessus, qui n'accroissent pas la redondance.

1. *Suppression d'une arête dans le dernier étage.* Si le dernier étage de l'arbre de code ne contient qu'une arête, alors à cet étage est lié le code élémentaire $B = B'b$ avec la probabilité p et, de plus, le mot B' n'est le préfixe d'aucun autre code élémentaire. Si l'on supprime cette arête et que l'on transfère la probabilité p au sommet d'où est issue cette arête, on obtient un nouvel arbre de code. A cette transformation correspond le passage du schéma Σ au schéma Σ' si l'on remplace dans Σ le code élémentaire B par B'. Il est évident que

$$l'_{\mathrm{m}} = l_{\mathrm{m}} - p \leqslant l_{\mathrm{m}}.$$

2. *Transport des arêtes du dernier étage vers un sommet non saturé de l'arbre de code.* Supposons que le dernier étage de l'arbre de code contienne au moins deux arêtes. Cela veut dire qu'il existe une arête au sommet terminal de laquelle est affectée la probabilité p $(p > 0)$, et un code élémentaire B. Supposons qu'il existe par ailleurs un sommet non saturé situé dans le l'-ième étage $(l' \leqslant l - 1)$. Désignons par B^0 le mot correspondant à ce sommet. La non-saturation du sommet entraîne l'existence d'un symbole b_j pour lequel B^0b_j n'est le préfixe d'aucun code élémentaire. Dans ce cas, on peut transporter l'arête mentionnée du dernier étage vers le sommet non saturé dans la direction de j. Donc, en remplaçant le code élémentaire B par B^0b_j, on déduit à partir du schéma Σ le schéma Σ':

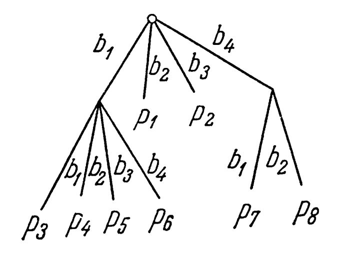


Fig. 84

$$l_{\mathrm{m}} = l_{\mathrm{m}} - pl + p\,(l\,(B^0) + 1) \leqslant l_{\mathrm{m}}.$$

Les transformations 1 et 2 permettent de convertir tout code sans préfixe de la classe considérée, y compris un code à redondance minimale, en un code dont l'arbre est saturé sans pour cela accroître la redondance. Ce qui prouve le lemme.

Exemple 10. En utilisant les transformations 1 et 2 on peut transformer l'arbre de code de la figure 84 en un arbre saturé.

Calculons l'_{m} pour ce code:

$$l'_{\mathrm{m}} = 0{,}22 + 0{,}20 + 2\,(0{,}14 + 0{,}11 + 0{,}10 + 0{,}09 + \\ + 0{,}08 + 0{,}06) = 0{,}42 + 2\cdot 0{,}58 = 1{,}58.$$

La redondance de ce code est inférieure à la redondance initiale.

Remarque. Considérons un code à redondance minimale dont l'arbre est saturé. Prenons le faisceau d'arêtes du dernier étage, incidentes vers l'extérieur au sommet excepté (voir définition de la page 202). Si ce sommet n'existe pas on prendra un faisceau quelconque du dernier étage. Soit q_0 le nombre d'arêtes, $2 \leqslant q_0 \leqslant q$. Par permutation des codes élémentaires de longueur maximale on peut faire en sorte que les sommets terminaux du faisceau considéré soient affectés des probabilités

$$p_{r-q_0+1}, \ldots, p_r.$$

Le code obtenu est dit *réduit*. Il est évident que pour un code réduit les probabilités $p_{r-q_0+1}, \ldots, p_r$ se définissent de façon unique, car elles le sont par le paramètre q_0 qui se déduit de façon unique à partir de l'équation (voir page 202).

Dans l'exemple 9, si l'on fait $r = 8$, $q = 4$, on trouve

$$8 = 3t + q_0,$$

c'est-à-dire que $t = q_0 = 2$. Au faisceau considéré sont affectées les probabilités p_7 et p_8.

Théorème 7 (de réduction). *Soit donné un code sans préfixe arbitraire à redondance minimale, de paramètres (r, q) et de probabilités $p_1, \ldots, p_r$. Dans l'arbre de code correspondante, considérons le sommet dont les suivants qui sont en nombre q_0*

$$2 \leqslant q_0 \leqslant q$$

ne sont que des terminaux.

Désignons par $p_{i_1}, \ldots, p_{i_{q_0}}$ $(p_{i_1} \geqslant \ldots \geqslant p_{i_{q_0}})$ les probabilités affectées aux sommets terminaux du faisceau d'arêtes considéré. Si $r > q$, alors en éliminant ce faisceau d'arêtes et en affectant le sommet terminal ainsi obtenu de la probabilité

$$p = p_{i_1} + \ldots + p_{i_{q_0}},$$

on obtient un arbre de code qui correspond à un code à redondance minimale de paramètres (r', q), où $r' = r - q_0 + 1$ et de probabilités

$$p_1, \ldots, p_{i_1-1}, p_{i_1+1}, \ldots, p_{i_{q_0}-1}, p_{i_{q_0}+1}, \ldots, p_r, p$$

(pour $i_1 = 1$ ou $i_{q_0} = r$ cette suite commence par p_{i_1+1} ou bien s'achève par $p_{i_{q_0}-1}$).

Démonstration. En effet, si le code obtenu n'était pas à redondance minimale, on aurait pu construire un code à redondance moindre pour les paramètres (r, q) et les probabilités $p_1, \ldots, p_r$.

Ce théorème, combiné avec les lemmes précédents, nous donne un algorithme de construction des codes à redondance minimale.

Il repose sur l'application du théorème de réduction au code réduit de paramètres (r, q) et de probabilités $p_1, \ldots, p_r$ $(p_1 \geqslant \ldots \geqslant p_r)$. Pour faisceau d'arêtes on prend les q_0 $(2 \leqslant q_0 \leqslant q)$ arêtes terminales. Le paramètre q_0 est défini de façon unique par les données initiales. Les probabilités

$$p_{r-q_0+1}, \ldots, p_r$$

affectées aux sommets terminaux du faisceau sont également définies de façon unique par les données initiales, ce qui permet de trouver les paramètres du code, obtenus par réduction pour $r > q$. On obtient $r' = r - q_0 + 1 < r$, $q' = q$ et les probabilités $p_1, \ldots, p_{r-q_0}, p$, où $p = p_{r-q_0+1} + \ldots + p_r$.

Donc, l'application multiple de la réduction nous conduit à un problème dans lequel $r \leqslant q$ et qui admet une solution triviale si l'on prend des mots à une lettre pour codes élémentaires.

Voici quelques exemples.

Exemple 11. Soient $r = 4$, $q = 2$ et $p_1 = 0{,}40$, $p_2 = 0{,}25$, $p_3 = 0{,}20$, $p_4 = 0{,}15$. Prenons $\mathfrak{B} = \{0, 1\}$. Dans le cas d'un alphabet à deux lettres, on peut concevoir la construction du code de la manière suivante

					→ 0,60	0 ←
p_1	0,40 →		0,40		→ 0,40	1
			→ 0,35	0		
p_2	0,25 →		0,25	1		
p_3	0,20	0				
p_4	0,15	1				

Nous avons une réduction en trois pas. Les termes regroupables sont réunis par des crochets. Pour construire les codes il faut choisir pour chaque crochet une correspondance biunivoque entre les probabilités et un sous-ensemble de $\mathfrak{B}$.

Dans notre cas associons 0 au nombre supérieur et 1 au nombre inférieur. Cheminons ensuite dans le sens contraire vers les symboles $p_1, \ldots, p_r$ et notons au passage des crochets le code correspondant. Par exemple le chemin

$$0{,}60 - 0{,}35 - 0{,}15 - p_4$$

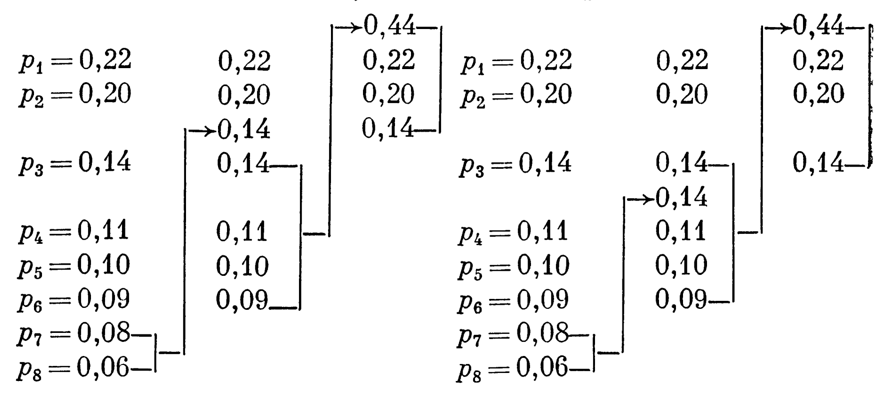
nous donne le code 001. On obtient ainsi le schéma suivant

a_1—1
a_2—01
a_3—000
a_4—001,

dans lequel on reconnaît le schéma Σ'' de la page 200. Donc le code défini par Σ'' est à redondance minimale.

Au cours de la réduction on classe à chaque pas les probabilités par ordre de grandeur. Cette mise en ordre n'est pas toujours unique, car on peut se trouver en présence de probabilités égales.

Exemple 12. Soient $r = 8$, $q = 4$ et $p_1 = 0{,}22$, $p_2 = 0{,}20$, $p_3 = 0{,}14$, $p_4 = 0{,}11$, $p_5 = 0{,}10$, $p_6 = 0{,}09$, $p_7 = 0{,}08$, $p_8 = 0{,}06$. Prenons $\mathfrak{B} = \{0, 1, 2, 3\}$. Il existe deux procédés de réduction:



De là on déduit deux schémas de codage alphabétique (en numérotant les termes entre crochets de haut en bas avec les nombres 0, 1, 2, 3):

	(Σ')		(Σ'')
a_1—1		a_1—1	
a_2—2		a_2—2	
a_3—00		a_3—3	
a_4—01		a_4—01	
a_5—02		a_5—02	
a_6—03		a_6—03	
a_7—30		a_7—000	
a_8—31		a_8—001	

L'arbre de code pour Σ' est confondu avec celui de la figure 84.

Considérons en conclusion la construction des codes à redondance minimale pour le cas où les probabilités $p_1, \ldots, p_r$ sont arbitraires, $p_1 \geqslant \ldots \geqslant p_r$. Si le nombre de probabilités nulles est plus grand que un, c'est-à-dire que $p_{r_0} > 0$ et

$$p_{r_0+1} = \ldots = p_r = 0, \quad r - r_0 > 1,$$

alors on trouve d'abord la solution pour l'alphabet d'entrée de $(r_0 + 1)$ lettres et pour les probabilités $p_1, \ldots, p_{r_0}, p_{r_0+1}$ $(p_{r_0+1} = 0)$. Supposons que $B_1, \ldots, B_{r_0}, B_{r_0+1}$ sont des codes élémentaires pour

le code à redondance minimale. On supprime ensuite le code élémentaire B_{r_0+1} et pour les lettres $a_{r_0+1}, \ldots, a_r$ on prend pour codes élémentaires, les mots de la forme

$$B'_{r_0+1} = B_{r_0+1}B^{(r_0+1)},$$
$$\cdots\cdots\cdots\cdots$$
$$B'_r = B_{r_0+1}B^{(r)},$$

où $l(B^{(r_0+1)}) = \ldots = l(B^{(r)})$ et tous les mots $B^{(r_0+1)}, \ldots, B^{(r)}$ sont distincts. Il est évident que le code construit est à redondance minimale et sans préfixe.

§ 36. Codes autocorrecteurs

Nous considérons ici un cas particulier de codage uniforme.

Soit $\mathfrak{A} = \{0, 1\}$ un alphabet contenant deux symboles. Soit d'autre part $\{A_1, A_2, \ldots, A_s\}$ l'ensemble de tous les mots $A = \alpha_1 \ldots \alpha_m$ formés à l'aide de l'alphabet $\mathfrak{A}$, de longueur fixe m ($s = 2^m$).

Supposons qu'un canal de transmission est perturbé par une source qui dans les mots de $\{A_1, \ldots, A_s\}$ de longueur m, déforme au plus p symboles. Cela veut dire que la séquence binaire obtenue à la sortie du canal se distingue de la séquence d'entrée en p positions au plus. Il est clair que si l'on transmet le message initial $\alpha_1 \ldots \alpha_m$ (sans le coder), il est impossible à la sortie du canal d'établir la nature du message envoyé. Il se pose ainsi la question de savoir s'il n'est pas possible de coder les mots A de l'ensemble $\{A_1, \ldots, A_s\}$, c'est-à-dire les mots de la forme $\alpha_1 \ldots \alpha_m$ par des mots $\beta_1 \ldots \beta_l$ de longueur l de telle sorte que d'après le code $\beta'_1 \ldots \beta'_l$ obtenu à la sortie lors de la transmission du code $\beta_1 \ldots \beta_l$ l'on puisse restituer ce code de façon unique et donc le message initial. Les codes jouissant de cette propriété sont dits *autocorrecteurs* par rapport à la source de perturbations considérée.

Il est immédiat de voir que ce problème admet une solution triviale. Nous allons le voir sur un exemple de source de perturbations élémentaire pour laquelle $p = 1$, c'est-à-dire pour laquelle ne sont possibles que deux perturbations: $0 \to 1$ ou $1 \to 0$. Le code autocorrecteur cherché s'obtient par triplement des symboles du code initial:

$$\alpha_1\alpha_2 \ldots \alpha_m - \alpha_1\alpha_1\alpha_1\alpha_2\alpha_2\alpha_2 \ldots \alpha_m\alpha_m\alpha_m.$$

En effet, si une erreur a été commise lors de la transmission de ce code, alors l'un seulement des trois symboles d'un groupe $\alpha_i\alpha_i\alpha_i$ est perturbé, les autres étant transmis fidèlement. Ceci permet par la méthode de « vote » de corriger les erreurs et de restituer le code $(\alpha_1\alpha_1\alpha_1 \ldots \alpha_m\alpha_m\alpha_m)$, donc le message initial $(\alpha_1 \ldots \alpha_m)$.

La solution triviale n'est pas correcte, car la longueur du code est égale à $l = 3m$ et nous sommes amenés à des codes pour lesquels la source de perturbations peut entraîner un nombre d'erreurs plus grand que p et alors il n'est pas toujours possible de restituer de façon unique le message initial.

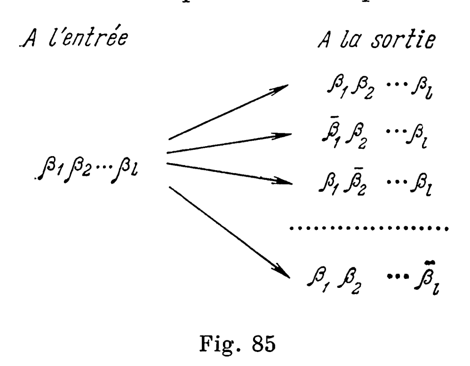


Fig. 85

On doit à Hamming la construction correcte de codes autocorrecteurs. Il a traité en détail le cas $p = 1$ à l'exposé duquel nous passons maintenant.

Les messages $\alpha_1 \ldots \alpha_m$ sont codés par les combinaisons $\beta_1 \ldots \beta_l$, où l est la longueur du code et $l = m + k$. Il est évident que cette source de perturbations donne lieu aux variantes suivantes (voir fig. 85).

Donc, le nombre de variantes est égal à $l + 1$. Pour que les positions supplémentaires du code $\beta_1 \ldots \beta_l$ suffisent pour coder les $l + 1$ cas énumérés de transmission du code, il est nécessaire que

$$2^k \geqslant l+1 \quad \text{ou} \quad 2^m \leqslant 2^l/(l+1).$$

Pour ces raisons on choisira l comme le plus petit entier vérifiant l'inégalité

$$2^m \leqslant 2^l/(l+1).$$

Les constructions ultérieures se font en trois étapes.

I. Construction des codes de Hamming (description de l'algorithme de codage). Partageons la tranche des entiers naturels $(1, 2, \ldots, l)$ en k suites de la manière suivante: soit V un entier naturel arbitraire $(1 \leqslant V \leqslant l)$ et $V_k \ldots V_1$ sa représentation binaire.

La suite 1, 3, 5, 7, 9, ... contient tous les nombres V dont $V_1 = 1$.

La suite 2, 3, 6, 7, 10, ... contient tous les nombres V dont $V_2 = 1$.

La suite 4, 5, 6, 7, 12, ... contient tous les nombres V dont $V_3 = 1$.

. . . .

La suite 2^{k-1}, $2^{k-1} + 1$, ... contient tous les nombres V dont $V_k = 1$.

Les premiers termes de ces suites sont

$$1 = 2^0,\ 2 = 2^1,\ \ldots,\ 2^{k-1},$$

c'est-à-dire les puissances de 2 et de plus $2^{k-1} \leqslant l$, et $2^k \geqslant l+1$.

Les termes β_i de la combinaison $\beta_1, \ldots, \beta_l$ dont l'indice i appartient à l'ensemble $(1, 2, \ldots, 2^{k-1})$ s'appellent *termes de contrôle*, les autres, *termes d'information*. Il est immédiat de voir que les termes de contrôle sont au nombre de k et ceux d'information, au nombre de $l - k = m$.

Formulons maintenant la règle de construction de la combinaison $\beta_1 \ldots \beta_l$ sachant la combinaison $\alpha_1 \ldots \alpha_m$. On commence par déterminer les termes d'information

$$\beta_3 = \alpha_1,$$
$$\beta_5 = \alpha_2,$$
$$\beta_6 = \alpha_3,$$
$$\ldots$$

Donc, la combinaison des termes d'information rangés par ordre de grandeur est confondue avec la combinaison $\alpha_1 \ldots \alpha_m$. On définit ensuite les termes de contrôle

$$\beta_1 = \beta_3 + \beta_5 + \beta_7 + \ldots \pmod{2},$$
$$\beta_2 = \beta_3 + \beta_6 + \beta_7 + \ldots \pmod{2},$$
$$\beta_4 = \beta_5 + \beta_6 + \beta_7 + \ldots \pmod{2},$$
$$\ldots\ldots\ldots\ldots\ldots$$

La sommation est effectuée sur les suites construites plus haut. Les seconds membres de ces formules sont visiblement constitués des termes d'information que nous avons déterminés plus haut. Désignons par H_l^1 l'ensemble de toutes les combinaisons $\beta_1 \ldots \beta_l$ construites.

II. Détection d'une erreur dans les codes de Hamming. Supposons que $(\beta_1 \ldots \beta_l) \in H_l^1$ et qu'une erreur a été commise dans le S-ième terme lors de la transmission du code $\beta_1 \ldots \beta_l$. Le mot reçu à la sortie du canal est $\beta_1' \ldots \beta_l'$, où

$$\beta_1' \ldots \beta_l' = \beta_1 \ldots \bar{\beta}_S \ldots \beta_l.$$

Soit $S_k \ldots S_1$ la représentation binaire de S. Montrons comment on peut trouver S sachant le code $\beta_1' \ldots \beta_l'$.

Considérons le nombre $S' = S_k' \ldots S_1'$, où

$$S_1' = \beta_1' + \beta_3' + \beta_5' + \beta_7' + \ldots \text{ (1-ière suite)},$$
$$S_2' = \beta_2' + \beta_3' + \beta_6' + \beta_7' + \ldots \text{ (2-ième suite)},$$
$$S_3' = \beta_4' + \beta_5' + \beta_6' + \beta_7' + \ldots \text{ (3-ième suite)},$$
$$\ldots\ldots\ldots\ldots\ldots$$

Tableau 41

1*	2*	3	4*	5	6	7	1*	2*	3	4*	5	6	7
0	0	0	0	0	0	0	1	1	1	0	0	0	0
1	1	0	1	0	0	1	0	0	1	1	0	0	1
0	1	0	1	$\bar{0}$	1	0	1	0	1	1	0	1	0
1	0	0	0	0	1	1	0	1	1	0	0	1	1
1	0	0	1	1	0	0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	1	0	1	0	1	0	1
1	1	0	0	1	1	0	0	0	1	0	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1

On affirme que $S = S'$. En effet, si $S_1 = 0$, alors S n'appartient pas à la 1-ière suite et

$$\beta_1' + \beta_3' + \beta_5' + \beta_7' + \ldots = \beta_1 + \beta_3 + \beta_5 + \beta_7 + \ldots = 0,$$

donc $S_1' = 0$; si $S_1 = 1$, alors S appartient à la 1-ière suite et

$$\beta_1' + \beta_3' + \beta_5' + \beta_7' + \ldots = 1 + \beta_1 + \beta_3 + \beta_5 + \beta_7 + \ldots = 1,$$

donc $S_1' = 1$. Donc $S_1 = S_1'$.

On démontre de façon analogue que $S_2 = S_2'$, ..., $S_k = S_k'$. D'où $S = S'$.

Si aucune erreur n'a été faite lors de la transmission, alors de toute évidence $S' = 0$. Donc, le nombre S' permet de dire si une erreur a eu lieu et alors de localiser le numéro du terme S perturbé et de le corriger en le remplaçant par $\overline{\beta_S'}$.

III. D é c o d a g e. Il consiste à restituer le message initial $\alpha_1 \ldots \alpha_m$ sachant $\beta_1 \ldots \beta_l$. Pour cela il suffit de toute évidence de prendre les termes d'information dans $\beta_1 \ldots \beta_l$.

Exemple 13. Construire un code autocorrecteur pour $m = 4$. Le plus petit nombre l vérifiant l'inégalité

$$2^4 \leqslant \frac{2^l}{l+1}$$

est $l = 7$. Donc $k = 3$. D'après l'étape I on obtient un code autocorrecteur qui est représenté dans le tableau 41. Les termes de contrôle sont suivis d'un astérisque.

Dans ce tableau on note d'abord dans les colonnes 3, 5, 6 et 7 (les termes d'information) de haut en bas les combinaisons 0000, ..., 1111. Ensuite on utilise les formules

$$\beta_1 = \beta_3 + \beta_5 + \beta_7 \ (\mathrm{mod.}\ 2),$$
$$\beta_2 = \beta_3 + \beta_6 + \beta_7 \ (\mathrm{mod.}\ 2),$$
$$\beta_4 = \beta_5 + \beta_6 + \beta_7 \ (\mathrm{mod.}\ 2)$$

pour remplir les colonnes 1, 2 et 4.

Supposons qu'à l'entrée du canal on applique le code 0110011 et que la source de perturbations déforme le 5-ième terme ($S = 5$). On obtient à la sortie 0110111. Calculons le numéro du terme corrompu. On obtient

$$S_1' = \beta_1' + \beta_3' + \beta_5' + \beta_7' = 0 + 1 + 1 + 1 = 1,$$
$$S_2' = \beta_2' + \beta_3' + \beta_6' + \beta_7' = 1 + 1 + 1 + 1 = 0,$$
$$S_3' = \beta_4' + \beta_5' + \beta_6' + \beta_7' = 0 + 1 + 1 + 1 = 1.$$

Donc, $S' = 101$, c'est-à-dire que $S' = 5$. Nous avons trouvé le terme corrompu et $S' = S$.

Etudions en conclusion quelques propriétés géométriques des codes de Hamming.

Traitons un cube unité à l dimensions comme un espace métrique sur lequel la distance $\rho(\beta', \beta'')$ de deux points $\beta' = (\beta_1', \ldots, \beta_l')$ et $\beta'' = (\beta_1'', \ldots, \beta_l'')$ est définie comme suit:

$$\rho(\beta', \beta'') = \sum_{i=1}^{l} |\beta_i' - \beta_i''|.$$

Il est manifeste que $\rho(\beta', \beta'')$ représente le nombre de coordonnées en lesquelles diffèrent les combinaisons β' et β''.

Théorème 8. *Pour toutes combinaisons* β' et β'' *telles que* $\beta' \neq \beta''$ *et* $\beta', \beta'' \in H_l^1$ *on a* $\rho(\beta', \beta'') \geqslant 3$.

Démonstration. Cette proposition sera démontrée si l'on exclut deux cas: a) $\rho(\beta', \beta'') = 1$; b) $\rho(\beta', \beta'') = 2$. En effet, si $\rho(\beta', \beta'') = 1$, une erreur sur l'unité est possible qui convertit le code β' en le code β''; si $\rho(\beta', \beta'') = 2$, il existe alors une combinaison β''' telle que $\rho(\beta', \beta''') = \rho(\beta''', \beta'') = 1$, c'est-à-dire que les codes β' et β'' peuvent se transformer en le code β''' si des erreurs sont commises sur les unités.

Donc, dans les deux cas on ne pourra dire lequel des codes β' ou β'' a été transmis, ce qui contredit le fait que le code H_l^1 est autocorrecteur. C.q.f.d.

Soit β^0 un point d'un cube unité à l dimensions.

Définition. On appelle *boule* de centre β^0 et de rayon p l'ensemble $U_l^p(\beta^0)$ des points β tels que $\rho(\beta^0, \beta) \leqslant p$.

Définition. On appelle *sphère* de centre β^0 et de rayon p l'ensemble $V_l^p(\beta^0)$ des points β tels que $\rho(\beta^0, \beta) = p$.

Il est évident que si le point β^0 est un point du « code », il se transformera, lors de la transmission par un canal perturbé par une source entraînant p erreurs au plus, en un point β tel que

$$\rho(\beta^0, \beta) \leqslant p,$$

c'est-à-dire que le point β appartiendra à la boule $U_l^p(\beta^0)$. D'où le

Théorème 9. *Pour qu'un ensemble H d'un cube unité à l dimensions soit un ensemble composé de codes autocorrecteurs pour une source de perturbations entraînant au plus p erreurs, il est nécessaire et suffisant que quels que soient* β' *et* β'' ($\beta' \neq \beta''$) *de H l'on ait*

$$\rho(\beta', \beta'') \geqslant 2p + 1.$$

Pour prouver ceci on remarquera simplement que la condition $\rho(\beta', \beta'') \geqslant 2p + 1$ revient à dire que les boules $U_l^p(\beta')$ et $U_l^p(\beta'')$ sont disjointes. Ceci veut dire à son tour qu'au code obtenu à la sortie du canal de transmission correspond un point qui appartient exactement à une boule.

Ce théorème nous fournit une approche géométrique pour la construction de codes autocorrecteurs. A ce théorème est rattachée la proposition suivante.

Théorème 10. a) *Pour* $l = 2^t - 1$ *le cube unité à l dimensions est une somme directe de boules unités.*

b) *Pour* $l = 2^t$ *le cube unité à l dimensions est une somme directe de sphères unités.*

Démonstration. a) Considérons un code de Hamming H_l^1 dans le cube unité à l dimensions, où $l = 2^t - 1$. Il est évident que

$$k = t, \quad m = 2^t - t - 1.$$

Considérons les boules unités centrées en les points de H_l^1. Montrons que ce système de boules définit la partition cherchée. Ces boules sont disjointes deux à deux. Donc le nombre total de points de ce système est égal à

$$(l+1)\,2^m = 2^t 2^{2^t - t - 1} = 2^{2^t - 1} = 2^l.$$

Donc, ce système de boules contient tous les points du cube unité à l dimensions.

b) Soit $l = 2^t$. Si la dernière coordonnée est fixe, le cube à l dimensions peut être « découpé » en deux cubes à $l - 1$ dimensions. Pour ces cubes il existe une correspondance biunivoque naturelle $\beta^0 \rightleftarrows \beta^1$, où

$$\beta^0 = (\beta_1, \ldots, \beta_{l-1}, 0), \quad \beta^1 = (\beta_1, \ldots, \beta_{l-1}, 1).$$

Comme $l - 1 = 2^t - 1$, chacun de ces cubes à $l - 1$ dimensions peut être, en vertu de a), représenté par une somme directe de boules unités. Choisissons dans l'un de ces cubes une partition en boules unités. On peut construire une partition dans l'autre en se servant de la correspondance biunivoque naturelle existant entre ces cubes.

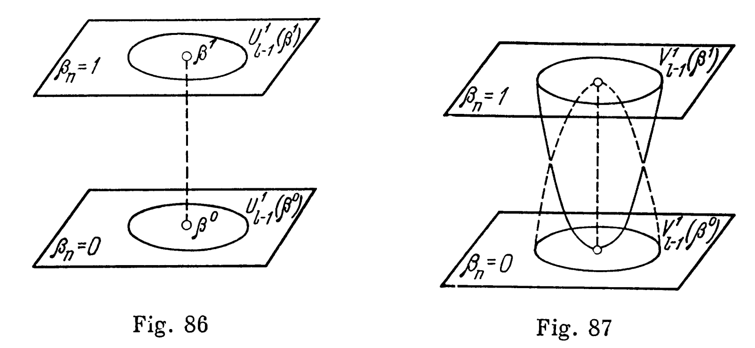


Fig. 86 Fig. 87

Considérons le couple de boules homologues (cf. fig. 86) $U^1_{l-1}(\beta^0)$ et $U^1_{l-1}(\beta^1)$.

Les deux boules unités à $l - 1$ dimensions peuvent être transformées en deux sphères unités à l dimensions (fig. 87).

En effet, chaque boule unité à $l - 1$ dimensions est la somme d'une sphère unité à $l - 1$ dimensions et de son centre, c'est-à-dire que

$$U^1_{l-1}(\beta^0) = V^1_{l-1}(\beta^0) \cup \{\beta^0\},$$
$$U^1_{l-1}(\beta^1) = V^1_{l-1}(\beta^1) \cup \{\beta^1\}.$$

Il est évident que pour les sphères unités à l dimensions on a

$$V^1_l(\beta^0) = V^1_{l-1}(\beta^0) \cup \{\beta^1\},$$
$$V^1_l(\beta^1) = V^1_{l-1}(\beta^1) \cup \{\beta^0\}.$$

Donc, si l'on effectue cette transformation pour tous les couples de boules homologues des partitions, on obtient la partition cherchée. C.q.f.d.

Définition. On dit qu'un ensemble de code appartenant à un cube unité à l dimensions et autocorrecteur pour la source de perturbations donnée est *maximal* si sa puissance est maximale.

Corollaire. *Le code de Hamming* H^1_l *est maximal pour* $l = 2^t - 1$.

QUATRIÈME PARTIE

QUELQUES APPLICATIONS À LA CYBERNÉTIQUE

CHAPITRE 7

FORMES NORMALES DISJONCTIVES

§ 37. Notion de forme normale disjonctive. Minimisation des fonctions booléennes

Soit donné un alphabet de variables $\{x_1, \ldots, x_n\}$.

Définition. Une expression de la forme

$$K = x_{i_1}^{\sigma_1} \& \ldots \& x_{i_r}^{\sigma_r} \ (i_\nu \neq i_\mu \text{ pour } \nu \neq \mu)$$

s'appelle *conjonction fondamentale*. Le nombre r s'appelle *rang* de la conjonction fondamentale. On conviendra que la constante 1 est une conjonction fondamentale de rang 0.

Définition. L'expression

$$\mathfrak{N} = \bigvee_{i=1}^{s} K_i \quad (K_i \neq K_j \text{ pour } i \neq j),$$

où K_i $(i = 1, \ldots, s)$ est une conjonction fondamentale de rang r_i, s'appelle *forme normale disjonctive*.

Il est évident que la forme normale disjonctive $\mathfrak{N}$ réalise une fonction booléenne $f(x_1, \ldots, x_n)$. Du chapitre 1 de la première partie il résulte que pour toute fonction $f(x_1, \ldots, x_n)$, $f \not\equiv 0$, il existe une forme normale disjonctive $\mathfrak{N}$ telle que

$$f(x_1, \ldots, x_n) = \mathfrak{N}.$$

Pour une telle forme $\mathfrak{N}$ on peut par exemple prendre la forme canonique disjonctive de f, c'est-à-dire

$$\mathfrak{N} = \bigvee_{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

Exemple 1. Considérons la fonction $f(x_1, x_2, x_3)$ définie par le tableau 45 (page 227). Cette fonction peut être représentée par la forme canonique disjonctive

$$\mathfrak{N}_1 = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3.$$

Par une vérification immédiate on s'assure d'autre part que cette fonction peut être représentée par la forme normale disjonctive

$$\mathfrak{N}_2 = \bar{x}_2\bar{x}_3 \vee x_1.$$

Cet exemple montre donc qu'une fonction booléenne admet plusieurs représentations par des formes normales disjonctives. D'où le problème de savoir laquelle de ces représentations est la plus payante. A cet effet on introduit un indice de simplicité $L(\mathfrak{N})$, qui caractérise le degré de complexité de la forme normale disjonctive.

On exigera de la fonctionnelle $L(\mathfrak{N})$ qu'elle satisfasse aux axiomes suivants.

I. *Axiome de positivité.* Pour toute forme normale disjonctive $L(\mathfrak{N}) \geqslant 0$.

II. *Axiome de monotonie* (pour la multiplication). Soit $\mathfrak{N} = \mathfrak{N}' \vee x_i^{\sigma_i} K'$. Alors

$$L(\mathfrak{N}) \geqslant L(\mathfrak{N}' \vee K').$$

III. *Axiome de convexité* (pour l'addition). Soit $\mathfrak{N} = \mathfrak{N}_1 \vee \mathfrak{N}_2$. Si $\mathfrak{N}_1 \& \mathfrak{N}_2 \equiv 0$, alors

$$L(\mathfrak{N}) \geqslant L(\mathfrak{N}_1) + L(\mathfrak{N}_2).$$

IV. *Axiome d'invariance* (pour l'isomorphisme). Supposons qu'une forme normale disjonctive $\mathfrak{N}'$ a été obtenue à partir d'une forme normale disjonctive $\mathfrak{N}$ par rebaptisation des variables (sans identification). Alors

$$L(\mathfrak{N}') = L(\mathfrak{N}).$$

Voici quelques exemples d'indices de simplicité pour formes normales disjonctives.

1. $L_L(\mathfrak{N})$ ou nombre des lettres-variables figurant dans l'écriture d'une forme normale disjonctive $\mathfrak{N}$. Pour les formes normales disjonctives $\mathfrak{N}_1$ et $\mathfrak{N}_2$ de l'exemple 1, $L_L(\mathfrak{N}_1) = 15$ et $L_L(\mathfrak{N}_2) = 3$, c'est-à-dire qu'au sens de cet indice la forme $\mathfrak{N}_2$ est plus simple que $\mathfrak{N}_1$.

2. $L_c(\mathfrak{N})$ ou nombre des conjonctions fondamentales entrant dans $\mathfrak{N}$. Pour les formes normales disjonctives $\mathfrak{N}_1$ et $\mathfrak{N}_2$ il est évident que $L_c(\mathfrak{N}_1) = 5$ et $L_c(\mathfrak{N}_2) = 2$, c'est-à-dire que la forme $\mathfrak{N}_2$ est plus simple que la forme $\mathfrak{N}_1$.

3. $L_0(\mathfrak{N})$ ou nombre de barres figurant dans $\mathfrak{N}$. Pour les formes $\mathfrak{N}_1$ et $\mathfrak{N}_2$ il est évident que $L_0(\mathfrak{N}_1) = 7$ et $L_0(\mathfrak{N}_2) = 2$, c'est-à-dire que la forme $\mathfrak{N}_2$ est plus simple que la forme $\mathfrak{N}_1$.

Il est immédiat de vérifier que chacun de ces indices satisfait aux axiomes indiqués.

Il est évident que sur l'alphabet de variables $\{x_1, \ldots, x_n\}$ on peut construire 3^n conjonctions fondamentales différentes (à la

conjonction « vide » est associée la constante 1). Il s'ensuit que le nombre de formes normales disjonctives sur cet alphabet de n lettres est égal à 2^{3^n}. Introduisons la définition suivante en nous appuyant sur ce calcul.

Définition. Une forme normale disjonctive $\mathfrak{N}$ réalisant une fonction $f(x_1, \ldots, x_n)$ et admettant un indice $L(\mathfrak{N})$ minimal s'appelle *forme normale disjonctive minimale pour L.*

Comme la suite de l'exposé est consacrée essentiellement à l'indice de simplicité L_L, toute forme normale disjonctive minimale pour cet indice sera simplement appelée *forme normale disjonctive minimale.* Une forme normale disjonctive minimale pour l'indice L_c sera dite *plus courte forme normale disjonctive.*

Revenons maintenant à l'exemple 1.

1\. La forme normale disjonctive $\mathfrak{N}_2 = \bar{x}_2\bar{x}_3 \vee x_1$ est minimale. En effet, la fonction $f(x_1, x_2, x_3)$ réalisée par cette forme dépend essentiellement des variables x_1, x_2 et x_3, donc ne peut être représentée par une forme normale disjonctive contenant moins de trois lettres.

2\. La forme normale disjonctive $\mathfrak{N}_2 = \bar{x}_2\bar{x}_3 \vee x_1$ est la plus courte, car la fonction $f(x_1, x_2, x_3)$ qu'elle réalise n'est pas une conjonction fondamentale.

3\. La forme normale disjonctive $\mathfrak{N}_2 = \bar{x}_2\bar{x}_3 \vee x_1$ est minimale pour L_0, car la fonction $f(x_1, x_2, x_3)$ qu'elle réalise par rapport aux variables x_2 et x_3 n'est pas croissante, et, ces deux variables étant essentielles, cette fonction ne peut être représentée par une forme normale disjonctive contenant moins de deux négations.

Notre but principal dans ce chapitre est de construire la forme normale disjonctive minimale pour L d'une fonction booléenne $f(x_1, \ldots, x_n)$. Ce problème s'appelle *problème de minimisation des fonctions booléennes.* Nous allons montrer qu'il admet une solution triviale. On commence par construire dans un ordre quelconque toutes les formes normales disjonctives (qui sont au nombre de 2^{3^n}) sur les variables $x_1, \ldots, x_n$

$$\mathfrak{N}_1, \mathfrak{N}_2, \ldots, \mathfrak{N}_{2^{3^n}}.$$

On sélectionne ensuite celles qui réalisent la fonction $f(x_1, \ldots, x_n)$. On calcule enfin les indices de simplicité des formes sélectionnées et par comparaison on trouve l'indice minimal. Cet algorithme est d'une réalisation très laborieuse car il implique l'examen de toutes les formes normales disjonctives, c'est-à-dire au moins 2^{3^n} opérations plus petites. On ne peut pratiquement pas l'utiliser à partir déjà de $n = 3$ (pour $n = 1$ et $n = 2$ le problème est trivial). Donc, il convient d'admettre que les algorithmes d'examen complet, c'est-à-

dire les algorithmes semblables par leur taille à l'algorithme trivial d'examen de toutes les formes normales disjonctives, sont interdits dans la résolution du problème de minimisation.

Attirons l'attention sur le fait que la théorie développée plus loin est valable pour tout indice de simplicité, donc l'étape initiale de minimisation est identique pour tous les indices de simplicité. D'autre part on peut admettre pour la commodité que dans ces constructions il est question de l'indice L_{L}, c'est-à-dire de la construction d'une forme normale disjonctive minimale. Comme une forme normale disjonctive minimale pour un indice est susceptible de ne pas l'être pour un autre *), la théorie élaborée pour un indice n'est en principe pas valable pour un autre. Cependant le fait que ces théories présentent pas mal d'affinités justifie l'étude de la minimisation pour un indice concret.

§ 38. Simplification des formes normales disjonctives et des formes normales disjonctives non redondantes

Soit $\mathfrak{N}$ une forme normale disjonctive et

$$\mathfrak{N} = \mathfrak{N}' \vee K, \quad \mathfrak{N} = \mathfrak{N}' \vee x_i^{\sigma_i} K',$$

où K est une conjonction fondamentale de $\mathfrak{N}$, $\mathfrak{N}'$ la forme normale disjonctive constituée des autres conjonctions de $\mathfrak{N}$, $x_i^{\sigma_i}$ un facteur de K, K' le produit des autres facteurs de K. Considérons deux types de transformations de formes normales disjonctives.

I. *Suppression d'une conjonction fondamentale.* On passe de la forme normale disjonctive $\mathfrak{N}$ à la forme $\mathfrak{N}'$ par suppression d'une conjonction fondamentale K. Cette transformation est définie si et seulement si $\mathfrak{N}' = \mathfrak{N}$.

II. *Suppression d'un facteur.* On passe de la forme normale disjonctive $\mathfrak{N}$ à la forme $\mathfrak{N}' \vee K'$ par suppression d'un facteur $x_i^{\sigma_i}$. Cette transformation est définie si et seulement si

$$\mathfrak{N}' \vee K' = \mathfrak{N}.$$

Définition. Une forme normale disjonctive $\mathfrak{N}$ qui ne peut être simplifiée par les transformations I et II s'appelle *forme normale disjonctive non redondante* (par ces transformations).

Exemple 2. Il est évident que la forme normale disjonctive $\mathfrak{N} = x_1 \vee \bar{x}_2 \bar{x}_3$ sera non redondante par les transformations I et II.

En se servant de ces transformations on peut formuler un algorithme de simplification d'une forme normale disjonctive. (Cet

*) Voir Vassiliev Y., *Problèmes de cybernétique*, vyp. 10, M.; Fizmatguiz, 1963, p. 5-61; *Mathématiques discrètes et problèmes mathématiques de cybernétique*, M.; Naouka, 1974, Livre 3.

algorithme est évident. Il est une variante de l'algorithme de plus forte pente, puisque $L(\mathfrak{N}') \leqslant L(\mathfrak{N})$ et $L(\mathfrak{N}' \vee K') \leqslant L(\mathfrak{N})$. Il est immédiat de voir que parmi les formes normales disjonctives non redondantes de la fonction $f(x_1, \ldots, x_n)$ il y en a toujours qui sont minimales.)

1. On choisit une forme normale disjonctive quelconque de la fonction $f(x_1, \ldots, x_n)$ en qualité de forme initiale. Pour une telle forme on peut, par exemple, prendre la forme canonique disjonctive, car elle est simple à construire.

2. Dans la forme initiale on ordonne les termes, puis les facteurs de chaque terme.

Tableau 42

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1	1	0	0	1
0	0	1	1	1	0	1	0
0	1	0	0	1	1	0	1
0	1	1	1	1	1	1	1

Exemple 3. Considérons la fonction $f(x_1, x_2, x_3)$ du tableau 42. Pour forme initiale prenons la forme canonique disjonctive et considérons deux ordres des termes : l'ordre naturel et un ordre spécial :

$$\mathfrak{N}' = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3,$$

$$\mathfrak{N}'' = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_3\bar{x}_1\bar{x}_2 \vee x_2\bar{x}_1x_3 \vee x_1x_2x_3 \vee \bar{x}_3x_1x_2 \vee x_1\bar{x}_2\bar{x}_3.$$

3. On procède ensuite à l'examen de la forme normale disjonctive (de la gauche vers la droite) ; on essaye d'abord de supprimer les conjonctions fondamentales K_i $(i = 1, \ldots, s)$; si cela est impossible, on considère les termes $x_{i_\nu}^{\sigma_\nu}$ de K_i de gauche à droite $(\nu = 1, \ldots$ $\ldots, r)$

$$K_i = x_{i_1}^{\sigma_1} \& \ldots \& x_{i_r}^{\sigma_r}$$

et on supprime le facteur $x_{i_\nu}^{\sigma_\nu}$ autant de fois qu'on le peut.

On passe ensuite à la conjonction fondamentale suivante.

Une fois qu'on a traité toutes les conjonctions fondamentales, on repasse en revue la forme normale disjonctive obtenue, de gauche à droite, en vue de supprimer éventuellement des conjonctions fondamentales *).

---

*) La nécessité de ce réexamen peut être illustrée par un exemple (voir tableau 44).

En définitive on obtient la forme normale disjonctive cherchée. On a manifestement le

Théorème 1. *La forme normale disjonctive obtenue par l'algorithme de simplification est une forme normale disjonctive non redondante (par les transformations* I *et* II).

Exemple 4. Soit la fonction $f(x_1, x_2, x_3)$ définie par le tableau 42. Prenons la forme canonique disjonctive pour forme initiale. Considérons l'ordre défini par l'écriture de $\mathfrak{N}'$ et étudions le fonctionnement de l'algorithme.

1. La conjonction $\bar{x}_1\bar{x}_2\bar{x}_3$ ne peut visiblement pas être supprimée. On peut néanmoins éliminer le facteur $\bar{x}_1$, puisque

$$\bar{x}_2\bar{x}_3 = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3.$$

On obtient la conjonction $\bar{x}_2\bar{x}_3$ d'où l'on ne peut éliminer aucun facteur.

2. On ne peut supprimer la conjonction $\bar{x}_1\bar{x}_2x_3$. Par contre, on voit qu'on peut y éliminer le facteur $\bar{x}_2$. On obtient donc la conjonction $\bar{x}_1x_3$ qui n'admet plus de simplification ultérieure.

3. La conjonction $\bar{x}_1x_2x_3$ peut être supprimée, puisque

$$\bar{x}_1x_3 = \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3.$$

4. Idem pour la conjonction $x_1\bar{x}_2\bar{x}_3$ (voir no 1).

5. La conjonction $x_1x_2\bar{x}_3$ ne peut être supprimée. Par contre, on peut y éliminer le facteur x_2. On obtient la conjonction non redondante $x_1\bar{x}_3$.

6. La conjonction $x_1x_2x_3$ ne peut visiblement pas être supprimée. On peut y éliminer le facteur x_1 et obtenir la conjonction x_2x_3.

On obtient la forme normale disjonctive

$$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1\bar{x}_3 \vee x_2x_3.$$

Le réexamen de cette forme ne laisse apparaître aucune simplification. Donc, l'algorithme de simplification nous conduit à la forme normale disjonctive $\mathfrak{N}_1$:

$$\mathfrak{N}_1 = \bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1\bar{x}_3 \vee x_2x_3.$$

Les calculs mentionnés peuvent être effectués comme l'indique le tableau 43.

Considérons toujours la même fonction et prenons l'ordre (spécial) de sa forme canonique disjonctive, soit

$$\mathfrak{N}'' = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_3\bar{x}_1\bar{x}_2 \vee x_2\bar{x}_1x_3 \vee x_1x_2x_3 \vee \bar{x}_3x_1x_2 \vee x_1\bar{x}_2\bar{x}_3.$$

Tableau 43

n° du pas	Forme normale disjonctive et ordre considéré	Conjonction étudiée	Nature de l'opération
1	$\bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee$ $\vee\ x_1x_2\bar{x}_3 \vee x_1x_2x_3$	$\bar{x}_1\bar{x}_2\bar{x}_3$	suppression de $\bar{x}_1$
2	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee$ $\vee\ x_1x_2\bar{x}_3 \vee x_1x_2x_3$	$\bar{x}_1\bar{x}_2x_3$	suppression de $\bar{x}_2$
3	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2\bar{x}_3 \vee$ $\vee\ x_1x_2\bar{x}_3 \vee x_1x_2x_3$	$\bar{x}_1x_2x_3$	suppression de $\bar{x}_1x_2x_3$
4	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3$	$x_1\bar{x}_2\bar{x}_3$	suppression de $x_1\bar{x}_2\bar{x}_3$
5	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1x_2\bar{x}_3 \vee x_1x_2x_3$	$x_1x_2\bar{x}_3$	suppression de x_2
6	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1\bar{x}_3 \vee x_1x_2x_3$	$x_1x_2x_3$	suppression de x_1
7	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1\bar{x}_3 \vee x_2x_3$ Le deuxième examen ne donne lieu à aucune simplification	Fin de l'algorithme	

Le tableau 44 donne les principales étapes du fonctionnement de l'algorithme correspondant à ce cas. On obtient la forme normale disjonctive

$$\mathfrak{N}_2 = \bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1x_2.$$

Cet exemple nous montre que le résultat de l'algorithme de simplification dépend du choix de l'ordre de la forme normale disjonctive initiale ; ainsi

$$L_{\mathrm{L}}(\mathfrak{N}_1) = 8, \quad L_{\mathrm{L}}(\mathfrak{N}_2) = 6$$

ou

$$L_{\mathrm{L}}(\mathfrak{N}_1) \neq L_{\mathrm{L}}(\mathfrak{N}_2).$$

Les formes normales disjonctives non redondantes peuvent être plus ou moins complexes et en particulier être différentes des formes minimales. D'où le problème de savoir s'il est possible de trouver pour toute fonction $f(x_1, \ldots, x_n)$, en partant d'un ordre des termes donné, une forme normale disjonctive minimale à l'aide de l'algorithme de simplification. La réponse est donnée par le théorème suivant.

Tableau 44

n° du pas	Forme normale disjonctive et ordre considéré	Conjonction étudiée	Nature de l'opération
	Premier examen de la forme normale disjonctive		
1	$\bar{x}_1\bar{x}_2\bar{x}_3 \vee x_3\bar{x}_1\bar{x}_2 \vee x_2\bar{x}_1x_3 \vee x_1x_2x_3 \vee \bar{x}_3x_1x_2 \vee x_1\bar{x}_2\bar{x}_3$	$\bar{x}_1\bar{x}_2\bar{x}_3$	suppression de $\bar{x}_1$
2	$\bar{x}_2\bar{x}_3 \vee x_3\bar{x}_1\bar{x}_2 \vee x_2\bar{x}_1x_3 \vee x_1x_2x_3 \vee \bar{x}_3x_1x_2 \vee x_1\bar{x}_2\bar{x}_3$	$x_3\bar{x}_1\bar{x}_2$	suppression de x_3
3	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2 \vee x_2\bar{x}_1x_3 \vee x_1x_2x_3 \vee \bar{x}_3x_1x_2 \vee x_1\bar{x}_2\bar{x}_3$	$x_2\bar{x}_1x_3$	suppression de x_2
4	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_1x_2x_3 \vee \bar{x}_3x_1x_2 \vee x_1\bar{x}_2\bar{x}_3$	$x_1x_2x_3$	suppression de x_1
5	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_2x_3 \vee \bar{x}_3x_1x_2 \vee x_1\bar{x}_2\bar{x}_3$	$\bar{x}_3x_1x_2$	suppression de $\bar{x}_3$
6	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_2 \vee x_1\bar{x}_2\bar{x}_3$	$x_1\bar{x}_2\bar{x}_3$	suppression de $x_1\bar{x}_2\bar{x}_3$
	Deuxième examen de la forme normale disjonctive		
7	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_3$	$\bar{x}_2\bar{x}_3$	irréductible
8	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_2$	$\bar{x}_1\bar{x}_2$	suppression de $\bar{x}_1\bar{x}_2$
9	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_2$	$\bar{x}_1x_3$	irréductible
10	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_2$	x_2x_3	suppression de x_2x_3
11	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1x_2$	x_1x_2	irréductible
12	$\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_3 \vee x_1x_2$	Fin de l'algorithme	

Théorème 2. *Soient* $f(x_1, \ldots, x_n)$ *une fonction booléenne quelconque* ($f \not\equiv 0$), $\mathfrak{N} = \bigvee_{i=1}^{s} K_i$ *une forme normale disjonctive non redondante* (*par les opérations* I *et* II) ; *alors pour un certain ordre des termes de la forme canonique disjonctive on peut obtenir la forme* $\mathfrak{N}$ *à l'aide d'un algorithme de simplification.*

Démonstration. Considérons la forme canonique disjonctive de la fonction $f(x_1, \ldots, x_n)$, ordonnée naturellement:

$$\mathfrak{N}^0 = \bigvee_{\substack{(\sigma_1, \ldots, \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

Soit $x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}$ un terme quelconque de cette forme. Comme $f(\sigma_1, \ldots, \sigma_n) = 1$, la forme normale disjonctive non redondante contient au moins une conjonction K_i, $i = i(\sigma_1, \ldots, \sigma_n)$, telle que

$$K_{i(\sigma_1, \ldots, \sigma_n)} = 1,$$

d'où il s'ensuit que $K_i = x_{i_1}^{\sigma_{i_1}} \& \ldots \& x_{i_r}^{\sigma_{i_r}}$. Ordonnons les facteurs du terme $x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}$ de telle sorte qu'en tête viennent les facteurs ne figurant pas dans K_i et ensuite pêle-mêle les facteurs de K_i. Donc

$$x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n} = K_\sigma \cdot K_{i(\sigma)} \quad (\sigma = (\sigma_1, \ldots, \sigma_n)).$$

On obtient ainsi une forme normale disjonctive $\mathfrak{N}'$. Il est immédiat de voir que l'algorithme de simplification de la forme normale disjonctive $\mathfrak{N}'$ conduit à l'un des résultats suivants: ou bien il supprime la conjonction $K_\sigma \cdot K_{i(\sigma)}$, ou bien il la transforme en la conjonction $K_{i(\sigma)}$. Donc, la forme normale disjonctive $\mathfrak{N}_1'$, résultat du fonctionnement de l'algorithme de simplification, est constituée uniquement des conjonctions fondamentales figurant dans $\mathfrak{N}'$. D'autre part, $\mathfrak{N}'$ étant non redondante, on a

$$\mathfrak{N}_1' = \mathfrak{N}'.$$

Corollaire. *Etant donné que parmi les formes normales disjonctives non redondantes il en existe nécessairement qui sont minimales pour L, l'algorithme de simplification permet de trouver une forme normale disjonctive minimale moyennant un choix convenable de l'ordre des termes de la forme canonique disjonctive.*

Remarque. Il ressort de la démonstration du théorème que pour construire des formes normales disjonctives non redondantes à l'aide de l'algorithme de simplification à partir d'une forme canonique disjonctive il suffit, si l'ordre des conjonctions est naturel, de faire varier simplement l'ordre des facteurs dans ces conjonctions.

Donc, pour construire une forme normale disjonctive minimale il faut considérer tous les ordres mentionnés des termes de la forme canonique disjonctive $\mathfrak{N}' = \bigvee_{i=1}^{s'} K_i'$ et pour chacun d'eux effectuer les calculs à l'aide de l'algorithme de simplification. Ceci permet d'évaluer la taille de cette procédure de minimisation.

De la définition on voit qu'une première application de l'algorithme de simplification est relativement simple et comprend $L_c(\mathfrak{N}')$

vérifications de la possibilité de supprimer une conjonction, $L_L(K'_i)$ vérifications de la possibilité de supprimer un facteur de K'_i ($i = 1, \ldots, s'$); la deuxième application comprend au plus $L_c(\mathfrak{N}')$ vérifications de la possibilité de suppression de conjonctions. Donc, le nombre total de vérifications

$$\sum_{i=1}^{s} L_L(K'_i) + 2L_c(\mathfrak{N}') \leqslant (n+2)\,2^n.$$

Le nombre total d'ordres des termes, ainsi qu'il résulte de la remarque, est égal à $(n!)^{s'}$ et *)

$$(n!)^{s'} \leqslant (n\,!)^{2^n} \leqslant \left(2\left(\frac{n}{2}\right)^n\right)^{2^n} = 2^{\left(n \log \frac{n}{2}+1\right)2^n}.$$

Donc, le nombre de vérifications pour tous les ordres n'est pas supérieur à

$$2^{\left(n \log \frac{n}{2}+1\right)2^n} \cdot (n+2)\,2^n,$$

nombre qui est de beaucoup inférieur à 2^{3^n}, c'est-à-dire que cet algorithme est meilleur que l'algorithme d'examen de toutes les formes normales disjonctives. Cependant, la taille de l'algorithme de simplification reste très grande.

Il reste encore à discuter un facteur influençant l'efficacité de l'algorithme: le choix de l'ordre des termes. Partageons les ordres mentionnés en « favorables » et « défavorables » selon que l'algorithme de simplification nous donne une forme normale disjonctive minimale ou non. Alors le rapport du nombre des ordres favorables au nombre de tous les ordres considérés est égal à la probabilité de construire une forme normale disjonctive minimale moyennant un choix aléatoire de l'ordre. On aurait pu évaluer ce rapport. Mais nous ne le ferons pas, car cette quantité est reliée à un algorithme concret et son estimation nous apprendra peu de choses sur la taille des autres algorithmes. Au lieu de cette quantité, on prendra pour mesure de la taille de l'algorithme une caractéristique similaire, notamment le rapport du nombre $\mu(f)$ des formes normales disjonctives minimales de la fonction f au nombre $\tau(f)$ de ses formes normales disjonctives non redondantes, c'est-à-dire le nombre $\mu(f)/\tau(f)$. Il s'avère qu'il existe des fonctions admettant un grand nombre de formes normales disjonctives non redondantes et un petit nombre de formes normales disjonctives minimales. On démontre qu'il existe une suite de fonctions $\{f_n\}$ pour laquelle

$$\mu(f_n)/\tau(f_n) \to 0.$$

---

*) Puisque $\sqrt{ab} \leqslant \frac{a+b}{2}$, on a $n! = n\,[((n-1)\cdot 1)\,((n-2)\cdot 2)\,\ldots] \leqslant 2\,\frac{n}{2}\left[\left(\frac{n}{2}\right)^2\left(\frac{n}{2}\right)^2\ldots\right] = 2\left(\frac{n}{2}\right)^n$.

Ceci nous incite à penser qu'il est peu probable que les raisonnements statistiques puissent être d'une quelconque utilité pour l'algorithme de simplification.

§ 39. Position du problème dans la forme géométrique

Désignons par E^n l'ensemble de toutes les combinaisons $\{\alpha_1, \ldots, \alpha_n\}$. Cet ensemble peut être traité comme celui des sommets d'un cube unité à n dimensions. Comme nous ne considérons

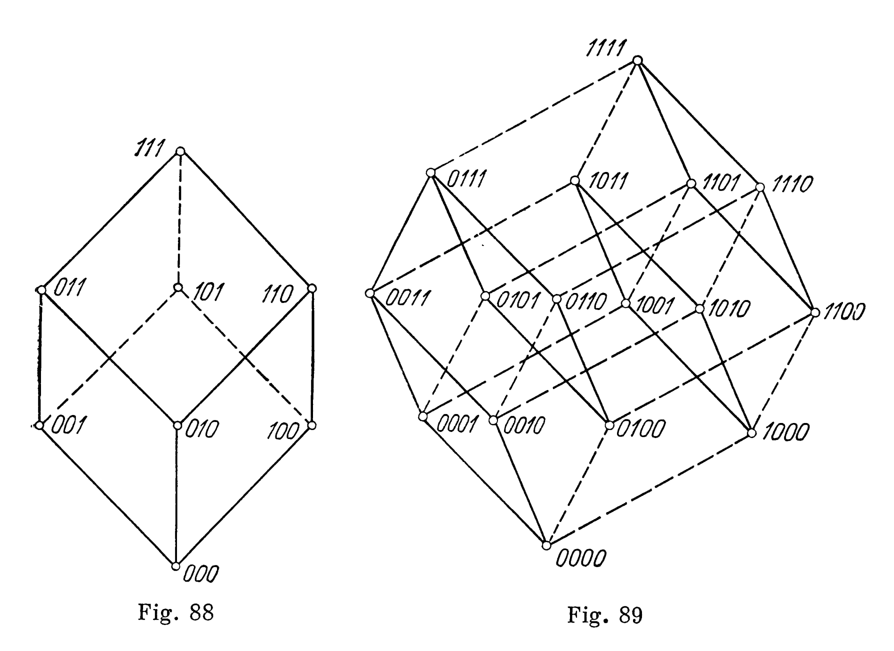


Fig. 88

Fig. 89

pas de points autres que ceux mentionnés, nous appellerons l'ensemble E^n cube à n dimensions et les combinaisons $\{\alpha_1, \ldots, \alpha_n\}$, sommets de ce cube. Les figures 88 à 91 représentent respectivement les projections de cubes à 3, 4, 5 et 6 dimensions sur le plan (sur la figure 91 les sommets ne sont pas des combinaisons mais les nombres naturels qui leur correspondent (voir page 10)).

Définition. Soit $\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_r}$ un système fixe de nombres 0 et 1, tel que $1 \leqslant i_1 < i_2 < \ldots < i_r \leqslant n$. L'ensemble de tous les sommets $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ du cube E^n tels que

$$\alpha_{i_1} = \sigma_{i_1}, \quad \alpha_{i_2} = \sigma_{i_2}, \ldots, \alpha_{i_r} = \sigma_{i_r}$$

s'appelle face à $n - r$ dimensions.

Il est évident qu'une face à $n - r$ dimensions est un sous-cube de E^n.

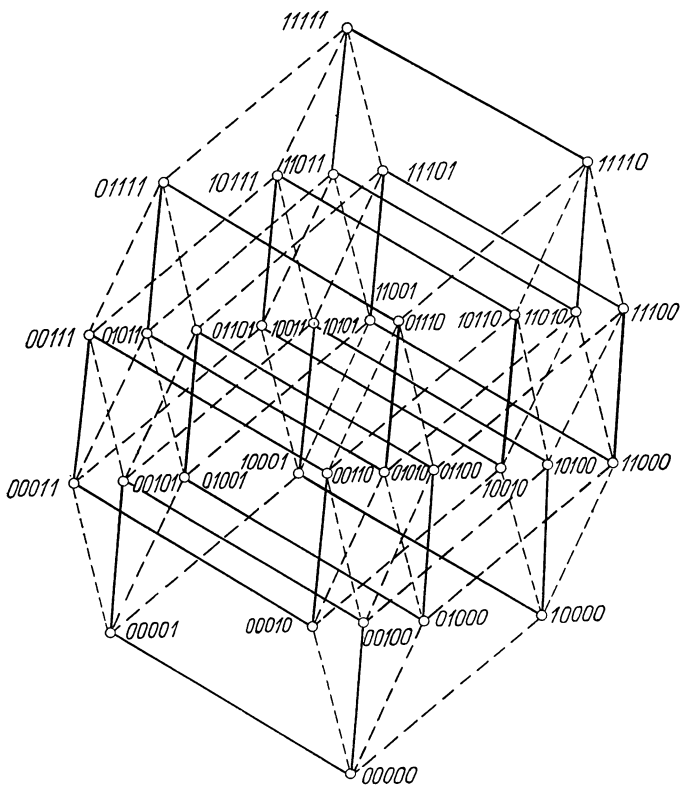


Fig. 90

Soit $f(x_1, x_2, \ldots, x_n)$ une fonction booléenne quelconque. Associons-lui un sous-ensemble N_f de sommets du cube E^n de telle sorte que

$$(\alpha_1, \alpha_2, \ldots, \alpha_n) \in N_f$$

si et seulement si

$$f(\alpha_1, \alpha_2, \ldots, \alpha_n) = 1.$$

Il est clair que la connaissance du sous-ensemble N_f nous permet de définir la fonction f de manière unique.

15—0382

Exemple 5. A la fonction $f(x_1, x_2, x_3)$ définie par le tableau 45 est associé l'ensemble

$N_f = \{(0, 0, 0), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}$,

représenté sur la figure 92.

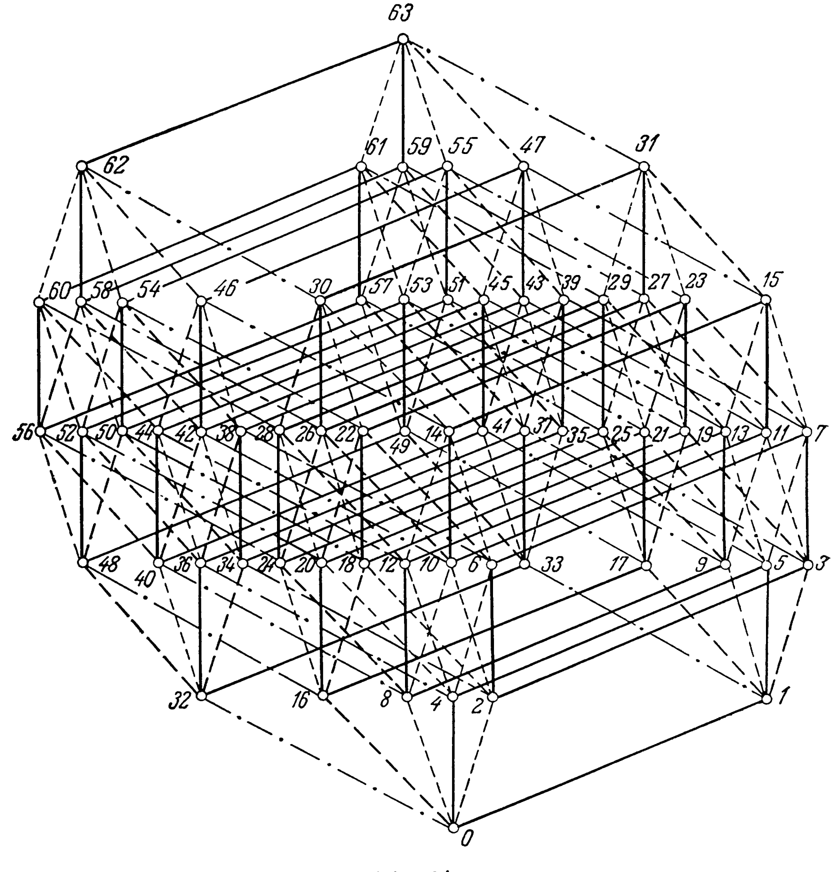


Fig. 91

Prenons pour fonction initiale une conjonction fondamentale $K(x_1, \ldots, x_n)$ de rang r, où

$$K(x_1, \ldots, x_n) = x_{i_1}^{\sigma_1} \& \ldots \& x_{i_r}^{\sigma_r}.$$

Définition. L'ensemble N_K correspondant à la conjonction K s'appelle *intervalle de rang* r.

Il est immédiat de voir que l'intervalle N_K de rang r est une face à $n-r$ dimensions.

Exemple 6. Aux conjonctions

$$K_1(x_1, x_2, x_3) = \bar{x}_2 \& \bar{x}_3,$$
$$K_2(x_1, x_2, x_3) = x_1 \& \bar{x}_2,$$
$$K_3(x_1, x_2, x_3) = x_1$$

sont associés les intervalles

$$N_{K_1} = \{(0, 0, 0), (1, 0, 0)\},$$
$$N_{K_2} = \{(1, 0, 0), (1, 0, 1)\},$$
$$N_{K_3} = \{(1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}$$

respectivement de rang 2, 2 et 1. Ces intervalles sont (voir fig. 92) des faces respectivement à une dimension (arête), à une dimension (arête) et à deux dimensions.

Signalons les propriétés évidentes de la correspondance introduite $f \rightleftarrows N_f$.

Si

$$f(x_1, \ldots, x_n) =$$
$$= g(x_1, \ldots, x_n) \vee h(x_1, \ldots, x_n),$$

alors

1) $N_g \subseteq N_f, \quad N_h \subseteq N_f$;

2) $N_f = N_g \cup N_h$.

Fig. 92

En particulier, si la fonction $f(x_1, \ldots, x_n)$ se représente par la orme normale disjonctive

$$\mathfrak{N} = K_1 \vee \ldots \vee K_s$$

alors de ces propriétés il s'ensuit que

$$N_{K_i} \subseteq N_f \quad (i = 1, 2, \ldots, s),$$

Tableau 45

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1	1	0	0	1
0	0	1	0	1	0	1	1
0	1	0	0	1	1	0	1
0	1	1	0	1	1	1	1

c'est-à-dire que l'image d'une conjonction K_i appartenant à la forme normale disjonctive de la fonction f est un intervalle situé à l'intérieur de l'ensemble N_f et

$$N_f = N_{K_1} \cup N_{K_2} \cup \ldots \cup N_{K_s},$$

c'est-à-dire qu'à la forme normale disjonctive de la fonction f correspond un recouvrement de l'ensemble N_f par les intervalles $N_{K_1}, \ldots, N_{K_s}$.

Il est immédiat de voir que la réciproque est également vraie: à tout recouvrement de l'ensemble N_f par des intervalles situés à l'intérieur de l'ensemble N_f correspond une forme normale disjonctive $\mathfrak{N}$ de la fonction f.

Exemple 7. Nous avons vu que

$$\mathfrak{N}_1 = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2\bar{x}_3 \vee x_1\bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3,$$
$$\mathfrak{N}_2 = \bar{x}_2\bar{x}_3 \vee x_1$$

sont des formes normales disjonctives de la fonction $f(x_1, x_2, x_3)$ (voir exemple 1). A ces formes correspondent deux recouvrements de l'ensemble N_f:

$$N_f = N_{K_1} \cup N_{K_2} \cup N_{K_3} \cup N_{K_4} \cup N_{K_5},$$
$$N_f = N_{K_1^0} \cup N_{K_2^0},$$

où

$$N_{K_1} = \{(0, 0, 0)\}, \quad N_{K_2} = \{(1, 0, 0)\},$$
$$N_{K_3} = \{(1, 0, 1)\}, \quad N_{K_4} = \{(1, 1, 0)\}, \quad N_{K_5} = \{(1, 1, 1)\},$$
$$N_{K_1^0} = \{(0, 0, 0), (1, 0, 0)\},$$
$$N_{K_2^0} = \{(1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}.$$

L'un de ces recouvrements est ponctuel, l'autre formé d'une arête et d'une face à deux dimensions.

Soit r_i le rang de l'intervalle N_{K_i} (il est égal à celui de la conjonction K_i). Le nombre r, où

$$r = \sum_{i=1}^{s} r_i$$

sera appelé *rang du recouvrement*. Nous pouvons maintenant formuler le problème de minimisation d'une fonction booléenne sous la forme géométrique (problème de recouvrement): trouver pour un ensemble donné N_f un recouvrement par des intervalles de N_f, soit

$$N_f = N_{K_1} \cup N_{K_2} \cup \ldots \cup N_{K_s},$$

dont le rang r soit minimal.

On peut donc considérer que le problème de minimisation d'une fonction booléenne admet deux positions. L'une, analytique (initiale), l'autre géométrique (problème de recouvrement). On utilisera donc deux langages : le langage analytique et le langage géométrique. Dans bien des cas on se servira d'un langage combiné dans lequel par exemple les conjonctions seront appelées intervalles et les formes normales disjonctives, recouvrements.

§ 40. Forme normale disjonctive réduite

Définition. On dit qu'un intervalle N_K contenu dans N_f est *maximal* (par rapport à N_f) s'il n'existe pas d'intervalle $N_{K'}$ tel que :

1. $N_K \subseteq N_{K'} \subseteq N_f$;
2. Le rang de l'intervalle $N_{K'}$ soit inférieur à celui de l'intervalle N_K.

Exemple 8. Soient

$$K_1(x_1, x_2, x_3) = \bar{x}_2 \& \bar{x}_3,$$

$$K_2(x_1, x_2, x_3) = x_1 \& \bar{x}_2,$$

$$K_3(x_1, x_2, x_3) = x_1.$$

Alors les intervalles N_{K_1} et N_{K_3} (voir fig. 92) sont maximaux, N_{K_2} ne l'est pas par rapport à N_f, puisque $N_{K_2} \subset N_{K_3}$ et le rang de N_{K_3} est inférieur à celui de N_{K_2}.

Définition. La conjonction K correspondant à un intervalle maximal N_K de l'ensemble N_f s'appelle *implicant simple de la fonction f*.

Comme la condition $N_K \subseteq N_{K'}$ équivaut au fait que tous les facteurs de K' sont contenus dans K, alors, dans un certain sens, on ne peut éliminer aucun facteur de l'implicant simple K de la fonction f, sinon, on obtiendrait (après suppression d'un facteur) une conjonction K' telle que $N_{K'} \not\subseteq N_f$.

Signalons une proposition évidente : tout intervalle N_K, $N_K \subseteq \subseteq N_f$, peut être élargi à un intervalle maximal.

Soit

$$N_{K_1^0}, N_{K_2^0}, \ldots, N_{K_m^0},$$

la liste de tous les intervalles maximaux de l'ensemble N_f. Il est aisé de voir que

$$N_f = N_{K_1^0} \cup N_{K_2^0} \cup \ldots \cup N_{K_m^0},$$

puisque $N_{K_i^0} \subseteq N_f$ $(i = 1, \ldots, m)$ et chaque point de N_f appartient à un intervalle maximal. La dernière égalité équivaut à la suivante :

$$f = K_1^0 \vee K_2^0 \vee \ldots \vee K_m^0.$$

Définition. On appelle *forme normale disjonctive réduite* la forme qui est la disjonction de tous les implicants simples de la fonction f.

Donc

$$\mathfrak{N}_R = K_1^0 \vee K_2^0 \vee \ldots \vee K_m^0$$

est une forme normale disjonctive réduite de la fonction f. Il résulte des considérations précédentes que cette forme est définie de façon unique par la fonction et réalise la fonction f.

Exemple 9. Pour la fonction f de l'exemple 6 on a le recouvrement suivant composé de tous les intervalles maximaux:

$$N_f = N_{K_1} \vee N_{K_3}.$$

A ce recouvrement est associée la forme normale disjonctive réduite

$$\mathfrak{N}_R = \bar{x}_2 \& \bar{x}_3 \vee x_1.$$

L'approche géométrique fournit en même temps une méthode de construction de la forme normale disjonctive réduite. Cependant il est souhaitable aussi de connaître la solution analytique.

Algorithme de construction d'une forme normale disjonctive réduite. Considérons une forme normale conjonctive (par exemple une forme canonique conjonctive) d'une fonction $f(x_1, \ldots, x_n)$. Chassons les parenthèses, c'est-à-dire effectuons une transformation de la forme

$$\& \vee \rightarrow \vee \&.$$

Puis, dans l'expression obtenue, éliminons les termes nuls, les termes absorbables et les termes doubles, c'est-à-dire effectuons les transformations

$$K_1K_2 \vee K_1 = K_1,$$
$$K_1 \vee K_1 = K_1.$$

En définitive on obtient la forme normale disjonctive réduite.

Exemple 10. Considérons la fonction $f(x_1, x_2, x_3)$ définie par le tableau 42. Sa forme canonique conjonctive est

$$f(x_1, x_2, x_3) = (x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee \bar{x}_3).$$

Chassons les parenthèses et simplifions:

$$(x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee \bar{x}_3) = x_1\bar{x}_1 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_1x_2 \vee$$
$$\vee x_2\bar{x}_2 \vee x_2x_3 \vee x_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3 \vee x_3\bar{x}_3 = \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee$$
$$\vee x_1x_2 \vee x_2x_3 \vee x_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3.$$

La forme normale disjonctive réduite s'écrit

$$\mathfrak{N}_R = \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_2x_3 \vee x_1x_2 \vee x_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3.$$

Cette expression se déduit aisément par des considérations géométriques et correspond à un cycle d'arêtes (voir fig. 93).

Exemple 11. Soit la fonction $f(x_1, x_2, x_3, x_4, x_5)$:

$$f(\alpha_1, \ldots, \alpha_5) = \begin{cases} 1 \text{ pour } 1 \leqslant \alpha_1 + \ldots + \alpha_5 \leqslant 3, \\ 0 \text{ dans les autres cas.} \end{cases}$$

Sa forme canonique disjonctive est manifestement

$$\mathfrak{N} = \bigvee_{1 \leqslant \sigma_1 + \ldots + \sigma_5 \leqslant 3} x_1^{\sigma_1} \& \ldots \& x_5^{\sigma_5}$$

et $L_c(\mathfrak{N}) = 25$. A chaque face contenue dans N_f correspond une conjonction fondamentale comprenant au moins deux facteurs barrés

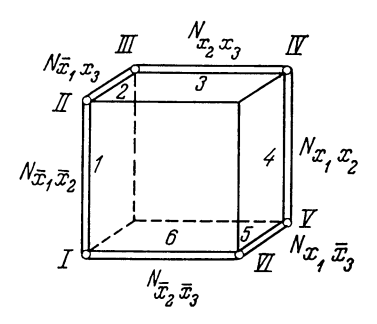


Fig. 93

et au moins un facteur non barré. Dans le même temps, à chaque conjonction de la forme $\bar{x}_i\bar{x}_jx_k$, où $i \neq j$, $i \neq k$ et $j \neq k$, correspond une face de l'ensemble N_f. D'où il résulte que toutes les conjonctions de la forme $\bar{x}_i\bar{x}_jx_k$ $(i \neq j,\ i \neq k,\ j \neq k)$ sont des implicants simples de la fonction f et

$$\mathfrak{N}_R = \bigvee_{\substack{(i,\ j,\ k) \\ i \neq j,\ i \neq k,\ j \neq k}} (\bar{x}_i\bar{x}_jx_k \vee \bar{x}_ix_j\bar{x}_k \vee x_i\bar{x}_j\bar{x}_k)$$

est la forme normale disjonctive réduite pour la fonction f. Il est évident que $L_c(\mathfrak{N}_R) = 3 \cdot C_5^3 = 30$.

Cet exemple montre que la forme normale disjonctive réduite de la fonction f peut contenir plus de termes que la forme canonique disjonctive.

§ 41. Non-redondance du point de vue géométrique. Méthodes de construction de formes normales disjonctives non redondantes

Définition. On dit qu'un recouvrement d'un ensemble N_f composé de faces maximales (par rapport à N_f) est *irréductible* si l'ensemble des faces obtenu par suppression de l'une d'elles n'est pas un recouvrement de N_f.

Définition. On appelle *non redondante* (au sens géométrique) la forme normale disjonctive correspondant à un recouvrement irréductible de l'ensemble N_f.

Exemple 12. Pour la fonction $f(x_1, x_2, x_3)$ définie par le tableau 42, on voit sur la figure 93 que

$$N_f = N_{\bar{x}_2\bar{x}_3} \cup N_{\bar{x}_1 x_3} \cup N_{x_1 x_2}$$

est un recouvrement irréductible et que

$$\mathfrak{N} = \bar{x}_2\bar{x}_3 \vee \bar{x}_1 x_3 \vee x_1 x_2$$

est une forme normale disjonctive non redondante (au sens géométrique).

Théorème 3. *Les notions de forme normale disjonctive non redondante par les transformations* I *et* II *et de forme normale disjonctive non redondante au sens géométrique sont équivalentes.*

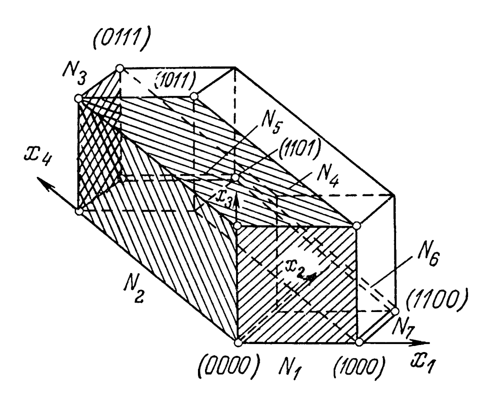


Fig. 94

Dans la suite on parlera simplement de formes normales disjonctives non redondantes sans mentionner la définition dont nous partons.

A noter que les formes normales disjonctives réduite, non redondante et minimale que nous avons définies sont reliées de la manière suivante.

Toute forme normale disjonctive non redondante se déduit de la forme réduite par suppression de certains termes.

Toute forme normale disjonctive minimale (par rapport à L_{L}) est non redondante.

Parmi les formes normales disjonctives non redondantes il en existe une qui est minimale (par rapport à L).

Considérons maintenant un exemple plus compliqué de construction de formes normales disjonctives non redondantes faisant appel à des raisonnements géométriques.

Exemple 13. Considérons la fonction $f(x_1, x_2, x_3, x_4)$ définie par le tableau 46. La figure 94 représente l'ensemble N_f. Cet ensemble comprend les faces maximales suivantes: les arêtes N_5, N_6, N_7 et les faces (à deux dimensions) N_1, N_2, N_3, N_4.

Tableau 46

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
0	1	0	0	0
0	1	1	0	0
1	0	0	1	0
1	1	1	0	0
1	1	1	1	0
sur les autres combinaisons				1

Donc, au recouvrement $N_1 \cup N_2 \cup N_3 \cup N_4 \cup N_5 \cup N_6 \cup N_7$ correspond la forme normale disjonctive réduite.

Les faces N_3 et N_4 appartiennent à n'importe quel recouvrement, car elles seules recouvrent les points (0111) et (1011). Le point (0000) peut être recouvert soit par la face N_1, soit par la face N_2.

a) Prenons la face N_1. Il reste à recouvrir les points (1100) et (1101). On peut le faire de deux manières: soit en prenant les arêtes N_5 et N_7, soit l'arête N_6. On obtient donc deux recouvrements irréductibles:

$$N_1 \cup N_3 \cup N_4 \cup N_5 \cup N_7, \quad N_1 \cup N_3 \cup N_4 \cup N_6.$$

b) Prenons la face N_2. Il reste à recouvrir les points (1000), (1100) et (1101). On peut le faire de deux façons: soit par les arêtes N_5 et N_7, soit par les arêtes N_6 et N_7. Ici aussi on obtient deux recouvrements irréductibles:

$$N_2 \cup N_3 \cup N_4 \cup N_5 \cup N_7, \quad N_2 \cup N_3 \cup N_4 \cup N_6 \cup N_7.$$

Pour construire les formes normales disjonctives non redondantes on remarquera qu'aux faces maximales $N_1, \ldots, N_7$ correspondent les implicants simples

$$K_1 = \bar{x}_2\bar{x}_4, \quad K_2 = \bar{x}_1\bar{x}_2, \quad K_3 = \bar{x}_1 x_4, \quad K_4 = \bar{x}_2 x_3,$$
$$K_5 = x_2\bar{x}_3 x_4, \quad K_6 = x_1 x_2\bar{x}_3, \quad K_7 = x_1\bar{x}_3\bar{x}_4.$$

On a

$$\mathfrak{N}_1 = \bar{x}_2\bar{x}_4 \vee \bar{x}_1 x_4 \vee \bar{x}_2 x_3 \vee x_2\bar{x}_3 x_4 \vee x_1\bar{x}_3\bar{x}_4,$$
$$\mathfrak{N}_2 = \bar{x}_2\bar{x}_4 \vee \bar{x}_1 x_4 \vee \bar{x}_2 x_3 \vee x_1 x_2\bar{x}_3,$$
$$\mathfrak{N}_3 = \bar{x}_1\bar{x}_2 \vee \bar{x}_1 x_4 \vee \bar{x}_2 x_3 \vee x_2\bar{x}_3 x_4 \vee x_1\bar{x}_3\bar{x}_4,$$
$$\mathfrak{N}_4 = \bar{x}_1\bar{x}_2 \vee \bar{x}_1 x_4 \vee \bar{x}_2 x_3 \vee x_1 x_2\bar{x}_3 \vee x_1\bar{x}_3\bar{x}_4\,;$$
$$L_{\mathrm{L}}(\mathfrak{N}_2) = 9, \quad L_{\mathrm{L}}(\mathfrak{N}_1) = L_{\mathrm{L}}(\mathfrak{N}_3) = L_{\mathrm{L}}(\mathfrak{N}_4) = 12.$$

Passons maintenant à la construction de l'algorithme qui permet de trouver toutes les formes normales disjonctives non redondantes à partir de notions géométriques.

Algorithme de construction des formes normales disjonctives non redondantes. Nous partons du recouvrement de l'ensemble N_f par le système de tous ses intervalles maximaux

$$N_{K_1^0}, \ldots, N_{K_m^0}.$$

Tableau 47

	P_0	P_1	...	P_j	...	P_λ
$N_{K_1^0}$	σ_{10}	σ_{11}	...	σ_{1j}	...	$\sigma_{1\lambda}$
...	...	...	...	...	...	...
$N_{K_i^0}$	σ_{i0}	σ_{i1}	...	σ_{ij}	...	$\sigma_{i\lambda}$
...	...	...	...	...	...	...
$N_{K_m^0}$	σ_{m0}	σ_{m1}	...	σ_{mj}	...	$\sigma_{m\lambda}$

Soient $N_f = \{P_1, \ldots, P_\lambda\}$ et P_0 un point tel que $P_0 \notin N_f$ (on admet que $f \not\equiv 1$). Dressons un tableau (voir tableau 47) dans lequel

$$\sigma_{ij} = \begin{cases} 0 & \text{si } P_j \notin N_{K_i^0} \quad (i = 1, \ldots, m), \\ 1 & \text{si } P_j \in N_{K_i^0} \quad (j = 0, 1, \ldots, \lambda). \end{cases}$$

Il est évident que la première colonne est composée de zéros, puisque $P_0 \notin N_f$, et que toute autre colonne contient au moins une unité. Donc, la première colonne est différente des autres.

Pour chaque j $(0 \leqslant j \leqslant \lambda)$ trouvons l'ensemble E_j de tous les numéros des lignes dans lesquelles la colonne P_j contient 1 (est différente de la colonne P_0).

Soit $E_j = \{e_{j1}, \ldots, e_{j\mu(j)}\}$. Formons l'expression

$$\underset{j=1}{\overset{\lambda}{\&}} (e_{j1} \vee \ldots \vee e_{j\mu(j)})$$

et effectuons la transformation

$$\& \vee \rightarrow \vee \&,$$

en traitant e comme des grandeurs booléennes. Dans l'expression obtenue supprimons les termes absorbables et les termes doubles, c'est-à-dire effectuons les transformations

$$A \cdot B \vee A = A,$$
$$A \vee A = A.$$

On obtient l'expression $\vee \&'$ qui est une partie de l'expression $\vee \&$. Il est évident que tout terme de $\vee \&'$ définira un recouvrement irréductible.

Exemple 14. Considérons la fonction $f(x_1, x_2, x_3)$ du tableau 42. Pour cette fonction l'ensemble N_f comprend 6 sommets numérotés I, II, . . ., VI. Les intervalles maximaux (voir fig. 93) sont les arêtes numérotées par les chiffres arabes. Dressons le tableau 48.

Tableau 48

	0	I	II	III	IV	V	VI
1	0	1	1	0	0	0	0
2	0	0	1	1	0	0	0
3	0	0	0	1	1	0	0
4	0	0	0	0	1	1	0
5	0	0	0	0	0	1	1
6	0	1	0	0	0	0	1

On a

$$E_{\mathrm{I}} = \{1, 6\}, \quad E_{\mathrm{II}} = \{1, 2\} \quad E_{\mathrm{III}} = \{2, 3\},$$
$$E_{\mathrm{IV}} = \{3, 4\}, \quad E_{\mathrm{V}} = \{4, 5\}, \quad E_{\mathrm{VI}} = \{5, 6\}.$$

Alors

$$\begin{aligned}\vee\& &= (1 \vee 6)(1 \vee 2)(2 \vee 3)(3 \vee 4)(4 \vee 5)(5 \vee 6) = \\ &= (1 \vee 2\cdot 6)(3 \vee 2\cdot 4)(5 \vee 4\cdot 6) = \\ &= (1\cdot 3 \vee 2\cdot 3\cdot 6 \vee 1\cdot 2\cdot 4 \vee 2\cdot 4\cdot 6)(5 \vee 4\cdot 6) = \\ &= 1\cdot 3\cdot 5 \vee 2\cdot 3\cdot 5\cdot 6 \vee 1\cdot 2\cdot 4\cdot 5 \vee \underline{2\cdot 4\cdot 5\cdot 6} \vee \\ &\quad \vee 1\cdot 3\cdot 4\cdot 6 \vee \underline{2\cdot 3\cdot 4\cdot 6} \vee \underline{1\cdot 2\cdot 4\cdot 6} \vee 2\cdot 4\cdot 6 = \\ &\quad = 1\cdot 3\cdot 5 \vee 2\cdot 3\cdot 5.6 \vee 1\cdot 2\cdot 4\cdot 5 \vee 1\cdot 3\cdot 4.6 \vee 2\cdot 4\cdot 6.\end{aligned}$$

On obtient cinq recouvrements irréductibles ou cinq formes normales disjonctives non redondantes:

$$\begin{aligned}\mathfrak{N}_1 &= \bar{x}_1\bar{x}_2 \vee x_2x_3 \vee x_1\bar{x}_3 \\ \mathfrak{N}_2 &= \bar{x}_1x_3 \vee x_2x_3 \vee x_1\bar{x}_3 \vee \bar{x}_2\bar{x}_3, \\ \mathfrak{N}_3 &= \bar{x}_1\bar{x}_2 \vee \bar{x}_1x_3 \vee x_1x_2 \vee x_1\bar{x}_3, \\ \mathfrak{N}_4 &= \bar{x}_1\bar{x}_2 \vee x_2x_3 \vee x_1x_2 \vee \bar{x}_2\bar{x}_3, \\ \mathfrak{N}_5 &= \bar{x}_1x_3 \vee x_1x_2 \vee \bar{x}_2\bar{x}_3.\end{aligned}$$

Deux d'entre elles, $\mathfrak{N}_1$ et $\mathfrak{N}_5$, sont minimales.

Cet algorithme est valable aussi pour l'exemple étudié à la page 233. Mais il peut être volumineux et pratiquement impropre pour les fonctions dépendant d'un nombre peu élevé de variables. Ceci tient à de multiples raisons: taille du tableau, complexité de la transformation $\&\vee \to \vee\&$, et en fin de compte au grand nombre de formes normales disjonctives non redondantes. Nous allons exhiber en conclusion un exemple de fonctions de quatre variables possédant un grand nombre de formes normales disjonctives non redondantes.

Exemple 15. Considérons la fonction

$$f(x_1, x_2, x_3, x_4) = \overline{x_1x_2x_3x_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4}.$$

Cette fonction est visiblement symétrique et sa forme normale conjonctive s'écrit

$$f(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee x_3 \vee x_4)(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4),$$

ce qui nous permet d'écrire immédiatement sa forme normale disjonctive réduite

$$\begin{aligned}\mathfrak{N}_R = x_1\bar{x}_2 \vee x_1\bar{x}_3 \vee x_1\bar{x}_4 \vee \bar{x}_1x_2 \vee x_2\bar{x}_3 \vee \\ \vee x_2\bar{x}_4 \vee \bar{x}_1x_3 \vee \bar{x}_2x_3 \vee x_3\bar{x}_4 \vee \bar{x}_1x_4 \vee \bar{x}_2x_4 \vee \bar{x}_3x_4.\end{aligned}$$

De là on voit que l'ensemble N_f possède 12 intervalles maximaux qui sont des faces à deux dimensions (cf. fig. 95). Les faces maximales forment une sphère. Numérotons ces faces comme sur la figure. Pour analyser les recouvrements irréductibles il est commode d'utiliser le développement de la sphère sur le plan (voir fig. 96). Dans ce développement les extrémités gauche et droite (en traitillé) doivent être collées et les segments verticaux (en traitillé) converger. Les faces 1 à 3 forment la bande supérieure, les faces 4 à 9, la bande médiane et les faces 10 à 12, la bande inférieure. Vient ensuite l'énumération des recouvrements irréductibles en fonction du fragment de ce recouvrement dans la bande médiane. Cette énumération envisage la possibilité de prolonger ce fragment dans les limites des bandes inférieure et supérieure. Il est particulièrement important de ne pas perdre de l'esprit les faces interdites (marquées du signe moins) et de recouvrir certains sommets (marqués d'un petit rond). Dans

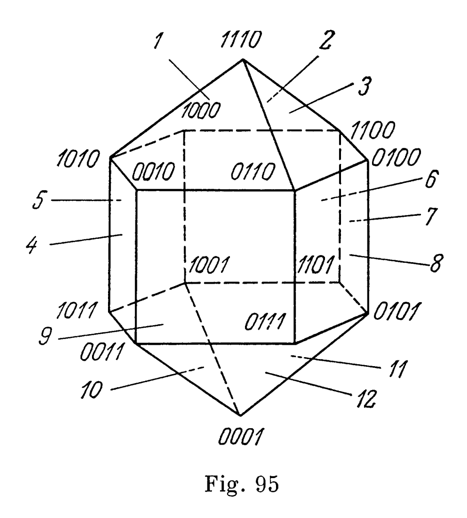


Fig. 95

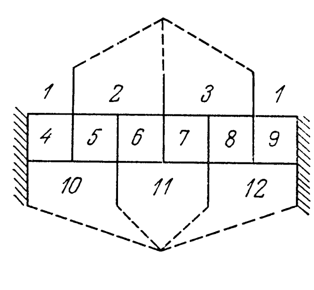


Fig. 96

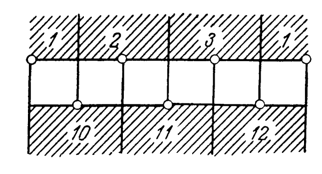


Fig. 97

chaque cas on calcule le nombre de variantes par la formule $a \cdot b \cdot c$, où a est le nombre de variantes dans la bande médiane, b dans la bande supérieure, c dans la bande inférieure.

1) Aucune face de la bande médiane n'est prise. Pour recouvrir les points marqués (voir fig. 97) il est nécessaire de prendre les bandes supérieure et inférieure tout entières $a \cdot b \cdot c = 1 \cdot 1 \cdot 1 = 1$.

2) La face 4 de la bande médiane est prise. Pour recouvrir les points marqués (voir fig. 98) il faut prendre les faces 2, 3 et 11, 12: $a \cdot b \cdot c = 6 \cdot 1 \cdot 1 = 6$.

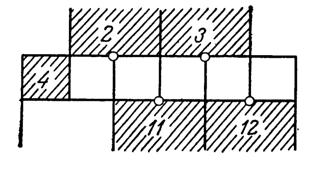


Fig. 98

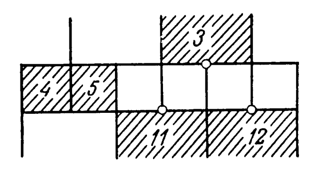


Fig. 99

3) On prend les faces voisines 4 et 5 de la bande médiane. Pour recouvrir les points marqués (voir fig. 99) il faut prendre les faces 3 et 11, 12: $a \cdot b \cdot c = 6 \cdot 1 \cdot 1 = 6$.

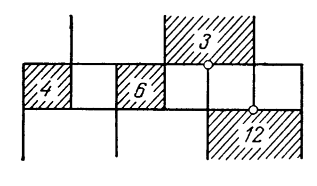


Fig. 100

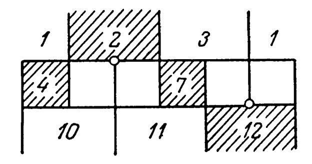


Fig. 101

4) On prend les faces 4 et 6 de la bande médiane. Pour recouvrir les points marqués (voir fig. 100) il faut prendre les faces 3 et 12: $a \cdot b \cdot c = 6 \cdot 1 \cdot 1 = 6$.

5) On prend les faces 4 et 7 de la bande médiane. Pour recouvrir les points marqués (voir fig. 101) il faut prendre les faces 2 et 12. Pour obtenir un recouvrement irréductible, il faut prendre l'une des

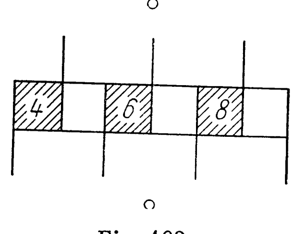


Fig. 102

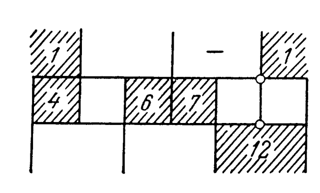


Fig. 103

faces 1 ou 3 de la bande supérieure. Si l'on opte pour la face 1, on ne peut ajouter que la face 11 de la bande inférieure, puisque la face 10 est interdite. Enfin si l'on choisit la face 3, on ne peut ajouter que la face 10 de la bande inférieure (la face 11 étant interdite): $a \cdot b \cdot c = 3 \cdot 2 \cdot 1 = 6$.

6) On prend les faces 4, 6 et 8 de la bande médiane. Pour recouvrir les points marqués (voir fig. 102) il faut prendre une face quelconque de la bande supérieure, par exemple la face 1. On peut alors prendre l'une quelconque des faces 11 et 12 de la bande inférieure (la face 10 est interdite): $a \cdot b \cdot c = 2 \cdot 3 \cdot 2 = 12$.

7-8) Dans la bande médiane on prend trois faces dont deux sont voisines et la troisième située à une face des deux premières (voir

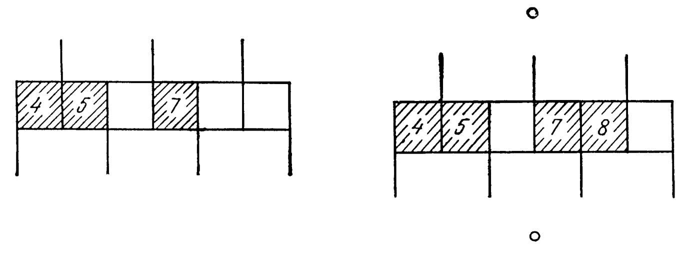


Fig. 104 Fig. 105

fig. 103 et 104). (Les deux variantes sont symétriques, donc il suffit d'étudier l'une d'elles.)

Pour recouvrir le point marqué (voir fig. 103) il faut prendre la face 12. Pour obtenir un recouvrement irréductible, il faut prendre encore une face de la bande supérieure, notamment la face 1 : $a \cdot b \cdot c = 6 \cdot 1 \cdot 1 = 6$.

9) Dans la bande médiane on prend quatre faces (non voisines trois à trois): les faces 4, 5 et 7, 8 (voir fig. 105).

Pour obtenir un recouvrement irréductible, il faut ajouter une face quelconque de la bande supérieure, par exemple, la face 1. On prend alors une face de la bande inférieure, la face 11, puisque les faces 10 et 12 sont interdites: $a \cdot b \cdot c = 3 \cdot 3 \cdot 1 = 9$.

On obtient en tout 58 recouvrements irréductibles, donc 58 formes normales disjonctives non redondantes dont 12 minimales.

§ 42. Quelques formes normales disjonctives obtenues de façon unique

La construction de formes normales disjonctives minimales à partir de la forme canonique disjonctive peut être caractérisée par le schéma suivant (voir fig. 106).

On obtient d'abord la forme normale disjonctive réduite (qui éventuellement peut être compliquée à cette étape). Ce processus « univoque » se transforme en un processus ramifié qui consiste à construire toutes les formes normales disjonctives non redondantes d'où l'on déterminera les formes minimales. L'étape la plus laborieuse de ce processus est la construction des formes

normales disjonctives non redondantes (partie branchue du processus). Il est toutefois possible d'alléger cette étape en procédant comme suit.

a) Eliminer préalablement les termes de la forme normale disjonctive réduite ne participant pas à la construction des formes normales disjonctives non redondantes, et réduire par là même l'examen

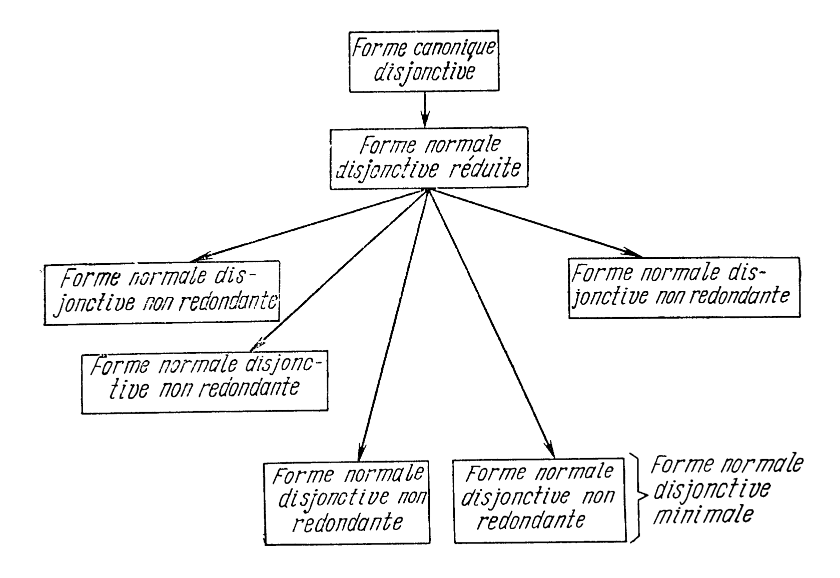


Fig. 106

(en considérant les sous-ensembles de la partie restante de la forme normale disjonctive réduite).

b) Eliminer une partie des termes de la forme normale disjonctive réduite de telle sorte que la partie restante permette de construire au moins une forme normale disjonctive minimale. Il est souhaitable que ce pas soit unique.

Dans ce paragraphe nous allons construire deux telles formes normales disjonctives définies de façon unique.

Définition. On dit qu'une face N_K, maximale par rapport à l'ensemble N_f, est *nucléaire* s'il existe un point $\tilde{\alpha}$ de N_f tel que $\tilde{\alpha} \in N_K$ et que $\tilde{\alpha}$ n'appartienne à aucune autre face maximale (par rapport à N_f).

Exemple 16. Considérons la fonction $f(x_1, x_2, x_3)$ du tableau 49. La figure 107 représente l'ensemble N_f et les faces maximales N_1, N_2 et N_3. Il est immédiat de voir que N_1 et N_3 sont des faces nucléaires,

puisque le point (0, 0, 0) est recouvert seulement par N_1 et le point (1, 1, 1), seulement par N_3.

Définition. L'ensemble de toutes les faces nucléaires pour N_f s'appelle *noyau.*

Dans l'exemple précédent $\{N_1, N_3\}$ est manifestement un noyau. Il est immédiat de voir que le noyau appartient à chaque recouvrement irréductible. D'où il résulte que les faces recouvertes par le noyau n'appartiennent à aucun recouvrement irréductible.

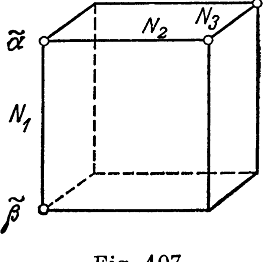


Fig. 107

Définition. On appelle *forme normale disjonctive de Quine* la forme normale disjonctive $\mathfrak{N}_Q$ obtenue à partir de la forme canonique disjonctive par suppression de tous les implicants simples correspondant aux faces maximales recouvertes par le noyau.

Théorème 4 (Quine [22]). *Toute fonction $f(x_1, \ldots, x_n)$ ($f \not\equiv 0$) admet une seule forme normale disjonctive de Quine.*

La définition de la forme normale disjonctive de Quine nous permet en fait de formuler l'algorithme de sa construction.

Tableau 49

x_1	x_2	x_3	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Exemple 17. Pour la fonction définie par le tableau 49 on a la forme normale disjonctive réduite

$$\mathfrak{N}_R = \bar{x}_1\bar{x}_2 \vee \bar{x}_2 x_3 \vee x_1 x_3.$$

Le noyau $\{N_1, N_3\}$ (voir page 241) recouvre la face N_2 à laquelle correspond l'implicant simple $\bar{x}_2 x_3$. La forme normale disjonctive de Quine s'écrit donc

$$\mathfrak{N}_Q = \bar{x}_1 \bar{x}_2 \vee x_1 x_3.$$

Ainsi, en supprimant quelques implicants de la forme normale disjonctive réduite on peut passer à une forme normale disjonctive définie de façon unique — la forme normale disjonctive de Quine — qui réalise la même fonction et contient toutes ses formes normales disjonctives non redondantes.

Le pas suivant de la partie « univoque » du processus de minimisation est lié aux formes normales disjonctives de type ΣT.

Définition. On appelle *forme normale disjonctive de type* ΣT la forme normale disjonctive notée $\mathfrak{N}_{\Sigma T}$, correspondant au recouvrement de l'ensemble N_f par l'ensemble de toutes les faces maximales (par rapport à N_f) qui appartiennent au moins à l'un des recouvrements irréductibles.

Il est évident que $\mathfrak{N}_{\Sigma T}$ s'obtient par la sommation logique (c'est-à-dire la disjonction) de toutes les formes normales disjonctives non redondantes de la fonction f et la réduction des termes semblables.

Il résulte de la définition que chaque fonction $f(x_1, \ldots, x_n)$ admet une forme normale disjonctive de type ΣT unique qui la réalise. Cette forme s'obtient à partir de la forme normale disjonctive réduite par suppression de certains termes.

Définition. Soit $\tilde{\alpha} \in N_f$. On appelle *faisceau passant par* $\tilde{\alpha}$ l'ensemble $\Phi_{\tilde{\alpha}}$ de toutes les faces maximales (par rapport à N_f) contenant le point $\tilde{\alpha}$.

Définition. Soient $\tilde{\alpha} \in N_f$ et N_{K^0} une face maximale telle que $\tilde{\alpha} \in N_{K^0}$. Alors le point $\tilde{\alpha}$ est dit *régulier* (*par rapport à* N_{K^0} *et* N_f) s'il existe un point $\tilde{\beta} \in N_f \setminus N_{K_0}$ et $\Phi_{\tilde{\beta}} \subseteq \Phi_{\tilde{\alpha}}$.

Exemple 18. Soit la fonction $f(x_1, x_2, x_3)$ du tableau 49 (voir aussi fig. 107). Prenons le sommet (0, 0, 1) pour point $\tilde{\alpha}$ et N_2 pour face maximale. Il est évident que $\tilde{\alpha} \in N_2$. Montrons que le point $\tilde{\alpha}$ est régulier (par rapport à N_2 et N_f). Soit $\tilde{\beta} = (0, 0, 0)$. On a : $\Phi_{\tilde{\alpha}} = \{N_1, N_2\}$, $\Phi_{\tilde{\beta}} = \{N_1\}$ et

$$\Phi_{\tilde{\beta}} \subseteq \Phi_{\tilde{\alpha}}.$$

Définition. On dit qu'une face N_{K^0} maximale pour N_f est *régulière* si chacun de ses points est régulier (par rapport à N_{K^0} et à N_f).

Dans notre exemple, il est évident que N_2 est une face régulière contrairement à N_1 et à N_3.

Théorème 5 (Y. Jouravliev [2]). *Pour qu'un implicant simple K^0 d'une fonction f n'appartienne pas à une forme normale disjonctive $\mathfrak{N}_{\Sigma T}$, il est nécessaire et suffisant que la face maximale correspondante N_{K^0} soit régulière.*

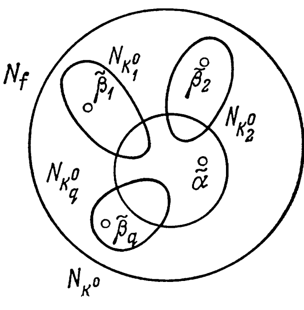


Fig. 108

Démonstration. *Nécessité.* Supposons que K^0 est un implicant simple de la fonction f, que K^0 n'appartient pas à une forme normale disjonctive de type ΣT et que par absurde N_{K^0} n'est pas une face régulière. Il existe alors un point $\tilde{\alpha} \in N_{K^0}$ qui n'est pas régulier. Soient $\tilde{\beta}_1, \ldots, \tilde{\beta}_q$ des points de l'ensemble $N_f \setminus N_{K^0}$ (voir fig. 108):

$$N_f \setminus N_{K^0} = \{\tilde{\beta}_1, \ldots, \tilde{\beta}_q\}.$$

Par hypothèse

$$\Phi_{\tilde{\beta}_1} \not\subseteq \Phi_{\tilde{\alpha}}, \ldots, \Phi_{\tilde{\beta}_q} \not\subseteq \Phi_{\tilde{\alpha}},$$

il existe donc des faces $N_{K_1^0}, \ldots, N_{K_q^0}$ appartenant respectivement aux faisceaux $\Phi_{\tilde{\beta}_1}, \ldots, \Phi_{\tilde{\beta}_q}$, telles que

$$\tilde{\alpha} \notin N_{K_1^0}, \ldots, \tilde{\alpha} \notin N_{K_q^0}.$$

De toute évidence

$$N_{K^0} \cup N_{K_1^0} \cup \ldots \cup N_{K_q^0} = N_f.$$

Ce recouvrement permet de construire un recouvrement irréductible de l'ensemble N_f. Ceci étant, certaines faces $N_{K_1^0}, \ldots, N_{K_q^0}$ risquent de disparaître, mais la face N_{K^0} figurera nécessairement dans le recouvrement irréductible, car elle seule recouvre le point $\tilde{\alpha}$. On obtient que K^0 figure dans une forme normale disjonctive non redondante, donc dans une forme de type ΣT. Cette contradiction prouve que N_{K^0} est régulière.

Suffisance. Supposons que N_{K^0} est une face régulière et montrons que K^0 n'appartient pas à une forme normale disjonctive de type ΣT.

Soient $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t$ des points de l'ensemble N_{K^0}, c'est-à-dire que

$$N_{K^0} = \{\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t\}.$$

La face N_{K^0} étant régulière, il existe des points $\tilde{\beta}_1, \ldots, \tilde{\beta}_t$ de $N_f \setminus N_{K^0}$ tels que

$$\Phi_{\tilde{\beta}_1} \subseteq \Phi_{\tilde{\alpha}_1}, \ldots, \Phi_{\tilde{\beta}_t} \subseteq \Phi_{\tilde{\alpha}_t}. \qquad (*)$$

Considérons une forme normale disjonctive non redondante $\mathfrak{N}$ de la fonction f et le recouvrement irréductible correspondant. Dans ce recouvrement

$$N_f = N_1 \cup \ldots \cup N_m,$$

il est évident que les points $\tilde{\beta}_1, \ldots, \tilde{\beta}_t$ sont recouverts respectivement par les faces $N_{i_1}, \ldots, N_{i_t}$ (voir fig. 109). En vertu des inclusions (*) ces faces recouvrent les points $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t$, c'est-à-dire que

$$N_{i_1} \cup \ldots \cup N_{i_t} \supseteq N_{K^0}.$$

Donc, la face N_{K^0} n'appartient pas au recouvrement irréductible considéré et l'implicant simple K^0 est une forme normale disjonctive $\mathfrak{N}$. Ceci achève la démonstration du théorème.

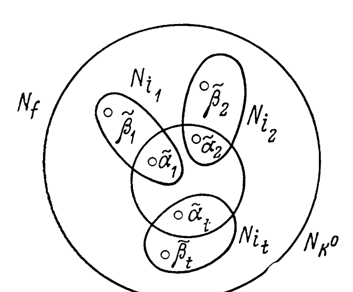


Fig. 109

Ce théorème nous donne le droit de formuler l'algorithme de construction d'une forme normale disjonctive de type ΣT: il faut éliminer de la forme normale disjonctive réduite toutes les conjonctions correspondant aux faces régulières.

Exemple 19. La fonction $f(x_1, x_2, x_3)$ du tableau 49 admet, d'après ce qui précède, une face régulière N_2. En l'éliminant, on obtient $N_1 \cup N_3$, ce qui nous donne la forme normale disjonctive $\mathfrak{N}_{\Sigma T}$ qui est confondue avec l'unique forme normale disjonctive non redondante de cette fonction.

Il se pose la question de savoir comment sont reliées les formes normales disjonctives de Quine et de type ΣT.

Théorème 6. *La forme normale disjonctive $\mathfrak{N}_{\Sigma T}$ d'une fonction f se déduit à partir de la forme normale disjonctive de Quine $\mathfrak{N}_Q$ de la même fonction par élimination éventuellement de quelques implicants simples.*

La démonstration résulte d'un fait évident : toute face maximale absorbée par un noyau est régulière.

L'exemple suivant montre que la forme normale disjonctive $\mathfrak{N}_{\Sigma T}$ peut être plus simple que la forme $\mathfrak{N}_Q$.

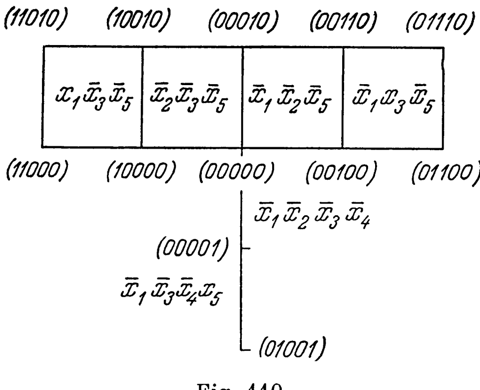


Fig. 110

Exemple 20. Considérons la fonction $f(x_1, x_2, x_3, x_4, x_5)$ définie par le tableau 50. La figure 110 représente la disposition des

Tableau 50

x_1	x_2	x_3	x_4	x_5	f	x_1	x_2	x_3	x_4	x_5	f
0	0	0	0	0	1	0	1	1	1	0	1
0	0	0	0	1	1	1	0	0	0	0	1
0	0	0	1	0	1	1	0	0	1	0	1
0	0	1	0	0	1	1	1	0	0	0	1
0	0	1	1	0	1	1	1	0	1	0	1
0	1	0	0	1	1	sur les autres combinaisons					0
0	1	1	0	0	1						

faces maximales pour N_f, soit quatre carrés et deux arêtes. Pour cette fonction, on a la forme normale disjonctive réduite

$$\mathfrak{N}_R = x_1\bar{x}_3\bar{x}_5 \vee \bar{x}_2\bar{x}_3\bar{x}_5 \vee \bar{x}_1\bar{x}_2\bar{x}_5 \vee \bar{x}_1 x_3\bar{x}_5 \vee \bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1\bar{x}_3\bar{x}_4 x_5.$$

L'arête $N_{\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4}$ est une face régulière contrairement aux autres, donc

$$\mathfrak{N}_{\Sigma T} = x_1\bar{x}_3\bar{x}_5 \vee \bar{x}_2\bar{x}_3\bar{x}_5 \vee \bar{x}_1\bar{x}_2\bar{x}_5 \vee \bar{x}_1 x_3\bar{x}_5 \vee \bar{x}_1\bar{x}_3\bar{x}_4 x_5.$$

D'autre part, comme le noyau constitué des arêtes

$$N_{x_1\bar{x}_3\bar{x}_5}, \quad N_{\bar{x}_1x_3\bar{x}_5}, \quad N_{\bar{x}_1\bar{x}_3\bar{x}_4x_5}$$

ne recouvre aucune face maximale ; on a

$$\mathfrak{N}_{\mathrm{Q}} = \mathfrak{N}_{\mathrm{R}} \neq \mathfrak{N}_{\Sigma\mathrm{T}}.$$

Donc, le processus de minimisation peut désormais être caractérisé par un schéma (voir fig. 111). Le processus branchu commence après la construction de la forme normale disjonctive $\mathfrak{N}_{\Sigma\mathrm{T}}$. Il est

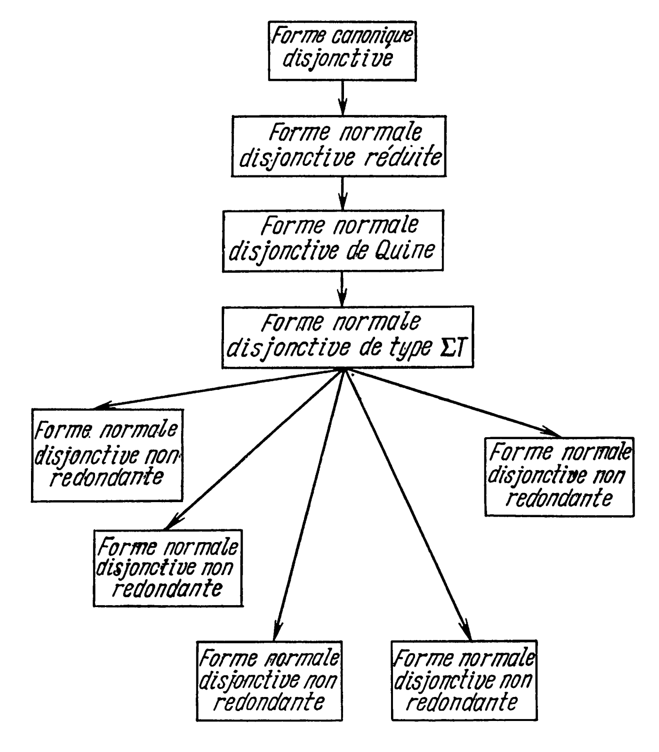


Fig. 111

possible certes de poursuivre la partie « univoque » du processus de minimisation. On peut par exemple pousser jusqu'à une forme normale disjonctive de type ΣM (la somme des formes normales disjonctives minimales avec réduction des termes semblables). Mais on verra plus bas que cela complique considérablement l'algorithme.

§ 43. Notion d'algorithme local

Les algorithmes de construction des formes normales disjonctives $\mathfrak{N}_Q$ et $\mathfrak{N}_{\Sigma T}$ sont spécifiques. Ils reposent sur l'étude spéciale du recouvrement de N_f par toutes les faces maximales, ce qui fournit une certaine information sur chaque face maximale. On voit par exemple si la face N_{K^0} est nucléaire (appartient à chaque recouvrement irréductible) ou non ; si elle est régulière (n'appartient à aucun recouvrement irréductible) ou non. On se sert ensuite de cette information pour éliminer certaines faces maximales: dans le premier cas les faces recouvertes par le noyau, dans le second, les faces régulières.

La notion de voisinage d'ordre u d'une face maximale N_{K^0} de l'ensemble N_f est essentielle pour la généralisation de ces algorithmes.

Définition (par récurrence). L'ensemble $S_0(N_{K^0}) = \{N_{K^0}\}$ s'appelle *voisinage d'ordre* 0 de la face maximale N_{K^0}. Soient définis les voisinages d'ordre $0, 1, \ldots, u-1$ de la face maximale N_{K^0}.

Alors le voisinage $S_u(N_{K^0})$ d'ordre u de la face maximale N_{K^0} se définit comme l'ensemble de toutes les faces maximales de N_f dont l'intersection avec $S_{u-1}(N_{K^0})$ n'est pas vide. Il est évident que

$$S_0(N_{K^0}) \subseteq S_1(N_{K^0}) \subseteq \ldots \subseteq S_4(N_{K^0}) \subseteq \ldots$$

Exemple 21. Prenons pour K^0 la conjonction $\bar{x}_1\bar{x}_2$ (voir fig. 93). Alors

$$S_0(N_{\bar{x}_1\bar{x}_2}) = \{N_{\bar{x}_1\bar{x}_2}\},$$
$$S_1(N_{\bar{x}_1\bar{x}_2}) = \{N_{\bar{x}_1\bar{x}_2}, N_{\bar{x}_2\bar{x}_3}, N_{\bar{x}_1x_3}\},$$
$$S_2(N_{\bar{x}_1\bar{x}_2}) = \{N_{\bar{x}_1\bar{x}_2}, N_{\bar{x}_2\bar{x}_3}, N_{\bar{x}_1x_3}, N_{x_1\bar{x}_3}, N_{x_2x_3}\},$$
$$S_3(N_{\bar{x}_1\bar{x}_2}) = \{N_{\bar{x}_1\bar{x}_2}, N_{\bar{x}_2\bar{x}_3}, N_{\bar{x}_1x_3}, N_{x_1\bar{x}_3}, N_{x_2x_3}, N_{x_1x_2}\},$$
$$S_u(N_{\bar{x}_1x_2}) = S_3(N_{\bar{x}_1\bar{x}_2}), \quad u \geqslant 3.$$

Ici

$$S_0(N_{\bar{x}_1\bar{x}_2}) \subset S_1(N_{\bar{x}_1\bar{x}_2}) \subset S_2(N_{\bar{x}_1\bar{x}_2}) \subset S_3(N_{\bar{x}_1\bar{x}_2}) = S_4(N_{\bar{x}_1\bar{x}_2}) = \ldots$$

Désignons par u_0 le plus petit ordre pour lequel

$$S_{u_0+1}(N_{K^0}) = S_{u_0}(N_{K^0}).$$

On voit sans peine (voir également exemple précédent) que

$$S_0(N_{K^0}) \subset S_1(N_{K^0}) \subset \ldots \subset S_{u_0}(N_{K^0}) = S_{u_0+1}(N_{K^0}) = \ldots,$$

et de plus chaque voisinage d'ordre u $(u > 0)$ contient au moins un point n'appartenant pas à un voisinage d'ordre $u-1$ si $u \leqslant u_0 - 1$. D'où $u_0 \leqslant 2^n$.

Fixons maintenant le paramètre u et étudions le recouvrement de l'ensemble N_f sur la base des informations recueillies sur ses voisinages d'ordre u. Cette étude rappelle l'établissement du plan d'une région sur la base des informations concernant des portions de cette région vues de points déterminés.

Nous allons voir un exemple qui explique le sens de l'étude locale des recouvrements. Soient

$$K_1^0 \vee \ldots \vee K_m^0$$

la forme normale disjonctive réduite d'une fonction f,

$$N_{K_1^0} \cup \ldots \cup N_{K_m^0}$$

un recouvrement (par les faces maximales) de l'ensemble N_f.

Définition. On dit que des faces $N_{K_i^0}$ et $N_{K_j^0}$ sont *connexes* s'il existe un u tel que

$$N_{K_j^0} \in S_u (N_{K_i^0}).$$

Il est aisé de voir que l'ensemble de toutes les faces $\{N_{K_i}\}$ se décompose de façon unique en composantes connexes, c'est-à-dire en sous-ensembles dans lesquels tout couple de faces est connexe, alors que les faces des sous-ensembles distincts ne le sont pas.

La question qui se pose est de savoir comment trouver la composante de connexité d'une face arbitraire N_{K^0}.

Repérons la conjonction K^0 et examinons les conjonctions dans l'ordre dans lequel elles se suivent dans la forme normale disjonctive réduite.

Si $N_{K^0} \in S_1 (N_{K_1^0})$, on repère la conjonction K_1^0. On passe ensuite à la conjonction K_2^0. On repère cette dernière si et seulement si $S_1(N_{K_2^0})$ contient des faces pour les conjonctions repérées, etc. L'examen de la forme normale disjonctive réduite nous permet de repérer certaines conjonctions. On commence ensuite un deuxième examen de la forme normale disjonctive réduite. Si l'ensemble des conjonctions repérées croît, on procède à l'examen suivant de la forme normale disjonctive réduite, etc. En définitive on arrive au cas (au bout d'un nombre d'examens inférieur à m) où l'ensemble des conjonctions repérées ne croît plus. On supprime alors toutes les conjonctions non repérées et le processus prend fin. L'ensemble restant de conjonctions n'est autre que la composante de connexité cherchée.

On remarque qu'on commence d'abord par étudier le recouvrement à l'aide de voisinages d'ordre un et ensuite sur la base de cette étude on repère les conjonctions. Ceci revient à dire que chaque conjonction est affectée d'une cellule de mémoire à deux états (« oui » et « non »). Il faut de plus une cellule pour indiquer si le nombre de conjonctions marquées croît ou non.

Pour deuxième paramètre prenons le nombre v des cellules admissibles de la mémoire binaire pour chaque conjonction. Fixons le paramètre v.

Passons maintenant à la description des algorithmes locaux *) sur les recouvrements par des faces maximales de paramètres u et v. Le fonctionnement de l'algorithme se décompose en deux étapes.

I. Etude du recouvrement. On étudie la forme normale disjonctive réduite dans un ordre donné et selon ce qui tombe dans le voisinage S_u (N_{K^0}) de la conjonction K^0, c'est-à-dire selon la partie de la forme normale disjonctive réduite et l'information recueillie sur les conjonctions de ce voisinage, on calcule les nouvelles valeurs $\gamma_1^0, \ldots, \gamma_v^0$ des cellules de la mémoire pour K^0. Ceci étant, on exige que les calculs soient conduits de la même manière pour deux formes normales disjonctives quelconques contenant la conjonction K^0 et dont les voisinages de la conjonction K^0 sont confondus avec les valeurs des cellules de la mémoire pour une conjonction du voisinage donné (caractère local de la mémorisation de l'information). Sous certaines conditions, le processus de mémorisation de l'information est « convergent » et peut donc s'achever. Dans ces cas on obtient une information finale définie par les valeurs des cellules de la mémoire pour les conjonctions de la forme normale disjonctive réduite.

II. Prise de la décision. Les valeurs calculées des cellules de la mémoire des conjonctions de S_u (N_{K^0}) déterminent la possibilité de supprimer la conjonction K^0. Cette procédure est locale aussi, c'est-à-dire qu'elle est identique pour deux formes normales disjonctives quelconques contenant K^0 et dont les voisinages de K^0 sont confondus avec les valeurs des cellules de la mémoire pour les conjonctions de ce voisinage.

Il est aisé de voir que l'algorithme de détermination de la composante de connexité d'une forme normale disjonctive réduite contenant la conjonction K^0 est un algorithme local de paramètres $u = 1$ et $v = 1$.

Dans la suite on admettra que f est telle que le recouvrement de l'ensemble N_f par l'ensemble de toutes ses faces maximales forme une composante connexe. Dans le cas contraire, le problème de minimisation à l'aide des algorithmes locaux se résout indépendamment pour chaque composante connexe.

Supposons que $f(x_1, \ldots, x_n)$ est douée de la propriété indiquée. Il est immédiat de voir qu'il existe un algorithme local de paramètres $u = 1$, $v = 2^n$ qui permet de construire une forme normale disjonctive minimale.

---

*) La notion d'algorithme local a été introduite par Y. Jouravliev (cf. DAN SSSR, **132**, n° 2, 1960). Nous donnons ici une variante légèrement différente de cette notion.

Première étape. Mettons en correspondance biunivoque les collections $\{\alpha_1, \ldots, \alpha_n\}$ et les nombres de l'ensemble $\{1, 2, \ldots, 2^n\}$:

$$(\alpha_1, \ldots, \alpha_n) \leftrightarrow i\,(\alpha_1, \ldots, \alpha_n).$$

A chaque conjonction K^0 de la forme normale disjonctive réduite de f associons le cortège binaire

$$(\gamma_1^0, \ldots, \gamma_{2^n}^0)$$

où $\gamma_{i(\alpha_1, \ldots, \alpha_n)}^0 = 1$ si et seulement si $K^0(\alpha_1, \ldots, \alpha_n) = 1$. Il est évident que $(\gamma_1^0, \ldots, \gamma_{2^n}^0)$ est un « code » pour la face K^0. On étudie ensuite le recouvrement à l'aide des voisinages d'ordre un et on calcule les nouvelles valeurs de $(\gamma_1^0, \ldots, \gamma_{2^n}^0)$ pour K^0 comme une disjonction des cortèges pour le voisinage considéré. En procédant ainsi on calcule pour chaque conjonction de la forme réduite un cortège binaire qui sera le même pour toutes les conjonctions, c'est-à-dire le « code » de la fonction (par exemple, une colonne du tableau définissant cette fonction).

Deuxième étape. Définissons la règle de décision de la manière suivante: considérons une forme normale disjonctive minimale quelconque de f et supprimons la conjonction envisagée si et seulement si elle ne figure pas dans ladite forme. Il est évident que cette règle est locale et nous conduit à une forme normale disjonctive minimale.

Cette circonstance signifie que dans les algorithmes locaux il est nécessaire d'imposer des restrictions à la taille de la mémoire admissible de v.

Montrons en conclusion de ce paragraphe que l'algorithme de Quine et l'algorithme de construction d'une forme normale disjonctive de type ΣT sont locaux et estimons leurs paramètres.

Dans le premier algorithme on étudie d'abord les conjonctions nucléaires. Pour cela il faut connaître les voisinages d'ordre un des conjonctions dans la forme normale disjonctive réduite et de se rappeler des repères: on prend le repère 0 si la conjonction n'est pas nucléaire et le repère 1 si elle l'est. Pour prendre la décision de supprimer une conjonction il faut connaître son voisinage d'ordre un et voir si elle est recouverte ou non par les conjonctions repérées de ce voisinage. On a donc un algorithme local de paramètres $u = 1$, $v = 1$.

Dans le second algorithme on s'assure d'abord si la conjonction considérée est régulière ou non. Pour cela on compare les faisceaux qui passent par les points de la face considérée (de la conjonction) et ceux qui passent par les points d'un voisinage d'ordre un de cette

face et qui ne sont pas contenus dans cette dernière, autrement dit il faut opérer sur les voisinages d'ordre deux. On affecte le symbole 1 à une face régulière et le symbole 0 à une face non régulière. Ensuite on prend la décision de supprimer les faces portant le symbole 1 (les faces régulières). Donc, on a affaire à un algorithme local de paramètres $u = 2$ et $v = 1$.

On constate que les algorithmes locaux englobent de nombreuses classes connues d'algorithmes. D'autre part, les algorithmes locaux de paramètres u et v sont des algorithmes à taille limitée.

CHAPITRE 8

SYNTHÈSE DES CIRCUITS D'ÉLÉMENTS FONCTIONNELS

Les *convertisseurs discrets*, c'est-à-dire des dispositifs (voir fig. 112) dotés d'un certain nombre d'entrées et de sorties tiennent une place importante dans la technique actuelle de calcul et de contrôle. Les signaux d'entrée et de sortie sont des éléments d'ensembles finis

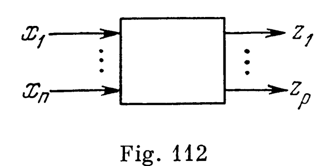


Fig. 112

connus. Ces dispositifs transforment les signaux d'entrée en signaux de sortie. Une sous-classe intéressante de convertisseurs discrets est celle des convertisseurs dans lesquels le temps de conversion est sensiblement inférieur à la durée des signaux (ou les convertisseurs dont on peut négliger le temps de conversion). Le modèle mathématique de ces dispositifs est constitué par les circuits d'éléments fonctionnels.

§ 44. Notion de circuit d'éléments fonctionnels

La notion de circuit d'éléments fonctionnels se définit en deux étapes. La première étape met en lumière la partie structurelle (diagrammique), la deuxième, la partie fonctionnelle de cette notion.

P r e m i è r e é t a p e. Définition d'un circuit d'éléments fonctionnels du point de vue de sa structure. Cette étape se décompose en plusieurs numéros.

1°. Soit un ensemble fini F d'objets F_i $(i = 1, \ldots, r)$ appelés *éléments*. Chaque élément F_i est doté de n_i entrées et d'une sortie. La figure 113 représente graphiquement un élément F_i.

2°. A l'aide de ces éléments nous allons donner par récurrence une définition (géométrique) de la notion de circuit logique. Par

circuit logique Σ on entendra un objet (cf. fig. 114) doté de n entrées et de p sorties.

a) Base de la récurrence. Un sommet isolé s'appelle *circuit logique* (*trivial*). Par définition, il est à la fois entrée et sortie (cf. fig. 115). Ici et dans la suite les flèches entrantes désigneront les entrées, les flèches partantes, les sorties.

b) Passage de la récurrence. Cette partie repose sur l'utilisation de trois opérations.

I°. Réunion des circuits disjoints. Soient Σ' et Σ'' deux circuits disjoints (sans éléments, entrées et sorties com-

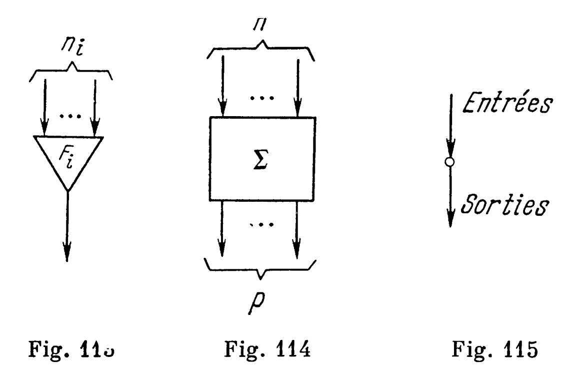


Fig. 113 Fig. 114 Fig. 115

muns) respectivement à n et m entrées et p et q sorties (cf. fig. 116). La réunion, au sens de la théorie des ensembles, des circuits logiques disjoints Σ' et Σ'' est un circuit logique Σ_1 doté de toutes les entrées

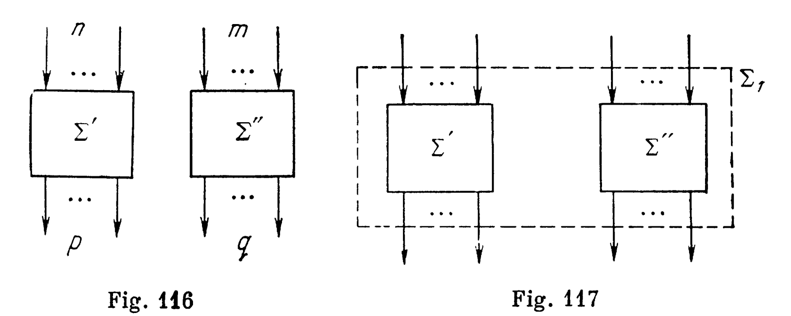


Fig. 116 Fig. 117

et de toutes les sorties des circuits Σ' et Σ''. Le circuit Σ_1 admet donc $n+m$ entrées et $p+q$ sorties (cf. fig. 117).

II°. Adjonction d'un élément F_i. Supposons qu'un circuit logique Σ' et un élément F_i (fig. 118) soient tels que $n_i \leqslant p$ et que dans Σ' on ait choisi n_i entrées deux à deux distinctes $j_1, j_2, \ldots$ $\ldots, j_{n_i}$. Alors la figure Σ_2 s'appelle circuit logique obtenu par connexion de l'élément F_i au circuit logique Σ'. Les entrées de Σ_2 sont

toutes celles de Σ', les sorties, toutes celles de Σ' excepté celles qui portent les numéros $j_1, \ldots, j_{n_i}$, plus la sortie de l'élément F_i. Le circuit logique Σ_2 admet n entrées et $p - n_i + 1$ sorties.

III°. Décomposition des sorties. Considérons la sortie j d'un circuit logique Σ' (fig. 119). La figure Σ_3 s'appelle circuit logique obtenu par décomposition de la sortie j. Les entrées

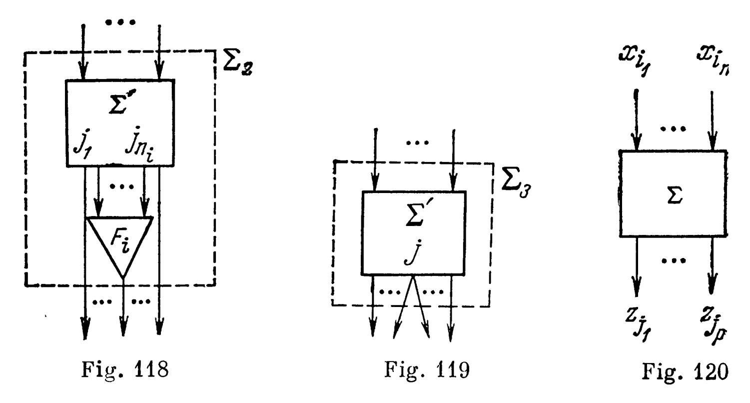


Fig. 118 Fig. 119 Fig. 120

de Σ_3 sont toutes celles de Σ', les sorties, toutes les sorties $1, \ldots, j-1, j+1, \ldots, p$ du circuit Σ' plus les deux sorties résultant de j. Donc, Σ_3 présente n entrées et $p+1$ sorties.

3°. Soient donnés les alphabets de variables $X = \{x_i\}$ et $Z = \{z_i\}$ *). Considérons un circuit logique Σ à n entrées et p sorties.

On appelle circuit d'éléments fonctionnels un circuit logique dont les entrées et sorties sont affectées de lettres différentes $x_{i_1}, \ldots$

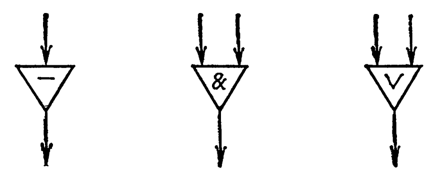


Fig. 121

$\ldots, x_{i_n}$ et $z_{j_1}, \ldots, z_{j_p}$ appartenant respectivement aux alphabets X et Z (voir fig. 120). Désignons le circuit obtenu par

$$\Sigma(x_{i_1}, \ldots, x_{i_n};\, z_{j_1}, \ldots, z_{j_p}).$$

Citons quelques exemples de circuits.

1. Supposons qu'un ensemble F est composé de trois éléments (cf. fig. 121).

*) Le symbole $\{x_i\}$ désigne l'ensemble de tous les x_i, où i parcourt l'ensemble (ou un sous-ensemble) des entiers naturels.

Alors le schéma Σ_1 de la figure 122 sera un circuit, puisqu'on peut le construire à partir de F à l'aide des opérations I°, II° et III°. Les étapes de la construction sont représentées sur le tableau 51.

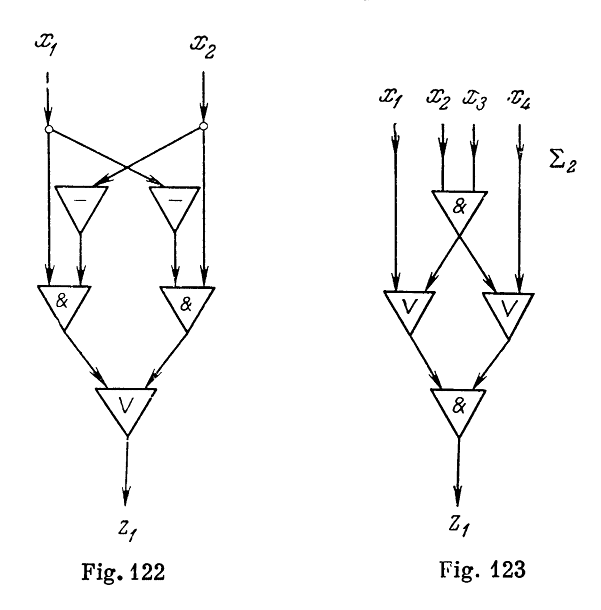


Fig. 122

Fig. 123

2. Le schéma Σ_2 de la figure 123 est également un circuit (voir tableau 52).

Dans la suite nous aurons affaire à des exemples de circuits dont l'entrée est à la fois sortie et présentant éventuellement plusieurs sorties.

D e u x i è m e é t a p e. Définition du fonctionnement d'un circuit.

4°. Soit $\Sigma(x_1, \ldots, x_n; z_1, \ldots, z_p)$ un circuit d'éléments fonctionnels *). Associons-lui un système d'équations (fonctions) booléennes

$$z_1 = f_1(x_1, \ldots, x_n),$$
$$\cdots\cdots\cdots\cdots$$
$$z_p = f_p(x_1, \ldots, x_n)$$

appelé aussi *conductibilité du circuit.* Pour cela à chaque élément F_i de l'ensemble F on associe un opérateur logique $f_i^0(\ldots, \ldots, \ldots)$ à n_i places défini par la fonction booléenne $f_i^0(y_1, \ldots, y_{n_i})$. On définit ensuite par récurrence la conductibilité du circuit.

*) Sans nuire à la généralité on peut admettre que les numéros des variables sont les premiers termes de la suite des entiers naturels.

Tableau 51

Circuit logique	Méthode d'obtention	Circuit logique	Méthode d'obtention
I	On prend deux circuits logiques triviaux. (Base de la récurrence)	II 1 2	On applique l'opération I à I
III 1 2 3 4	Dans II, on décompose les sorties 1 et 2. Opération III (2 fois)	IV 1 2 3 4	On branche les éléments «—» aux sorties 2 et 3 du circuit III. Opération II (2 fois)
V 1 2	On branche les éléments « & » aux sorties (1, 2) et (3, 4) du circuit IV. Opération II (2 fois)	VI x_1 x_2	On branche les éléments « $\vee$ » aux sorties 1 et 2 du circuit V. Opération II

a) B a s e d e l a r é c u r r e n c e. Le circuit Σ est un circuit trivial (cf. fig. 124). Dans ce cas l'équation est de la forme

$$z_1 = x_1$$

et la conductibilité est aussi une fonction identique.

b) P a s s a g e d e l a r é c u r r e n c e. Soient Σ' et Σ'' deux circuits disjoints aux entrées et sorties desquels sont affectées respectivement des lettres différentes (fig. 125). Supposons d'autre part qu'aux circuits

$$\Sigma'(x_1, \ldots, x_n; z_1, \ldots, z_p), \quad \Sigma''(x_{n+1}, \ldots, x_{n+m}; z_{p+1}, \ldots, z_{p+q})$$

sont associés les systèmes d'équations

$$\begin{matrix} z_1 = f_1(x_1, \ldots, x_n), \\ \cdots\cdots\cdots \\ z_p = f_p(x_1, \ldots, x_n), \end{matrix} \quad (*) \qquad \begin{matrix} z_{p+1} = f_{p+1}(x_{n+1}, \ldots, x_{n+m}), \\ \cdots\cdots\cdots \\ z_{p+q} = f_{p+q}(x_{n+1}, \ldots, x_{n+m}). \end{matrix} \quad (**)$$

Tableau 52

Circuit logique	Méthode d'obtention	Circuit logique	Méthode d'obtention
I 1 2 3 4	On prend quatre circuits logiques triviaux	II 1 2 3 4	On applique l'opération I (3 fois) à I
III & 1 2 3	On branche l'élément « & » aux sorties 2 et 3 du circuit II. Opération II	IV & 1 2 3 4	On décompose la sortie 2 dans III. Opération III
V & ∨ ∨ 1 2	On branche les éléments « ∨ » aux sorties (1, 2) et (3, 4) du circuit IV. Opération II (2 fois)	x_1 x_2 x_3 x_4 VI & ∨ ∨ & z_1	On branche l'élément « & » aux sorties (1, 2) du circuit V. Opération II

I. Au circuit $\Sigma_1(x_1, \ldots, x_{n+m}; z_1, \ldots, z_{p+q})$, réunion des circuits Σ' et Σ'', associons le système d'équations, réunion des équations (∗) et (∗∗):

$$z_1 = f_1'(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$z_p = f_p'(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}),$$
$$z_{p+1} = f_{p+1}'(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$z_{p+q} = f_{p+q}'(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}).$$

Ici $f_1', \ldots, f_p'$ ne dépendent pas essentiellement des variables $x_{n+1}, \ldots, x_{n+m}$, et $f_{p+1}', \ldots, f_{p+q}'$, des variables $x_1, \ldots, x_n$ et

$$f_1' = f_1, \ldots, f_p' = f_p, \quad f_{p+1}' = f_{p+1}, \ldots, f_{p+q}' = f_{p+q}.$$

17—0382

II. Supposons que le circuit

$$\Sigma_2(x_1, \ldots, x_n; z_1, \ldots, z_{j_1-1}, z_{j_1+1}, \ldots, z_{j_{n_i}-1}, z_{j_{n_i}+1}, \ldots, z_p, z_{p+1})$$

a été déduit à partir du circuit $\Sigma'(x_1, \ldots, x_n; z_1, \ldots, z_p)$ par adjonction aux sorties $z_{j_1}, \ldots, z_{j_{n_i}}$ $(n_i \leqslant p)$ de l'élément F_i et soit z_{p+1} la nouvelle sortie. Associons à ce circuit le système d'équa-

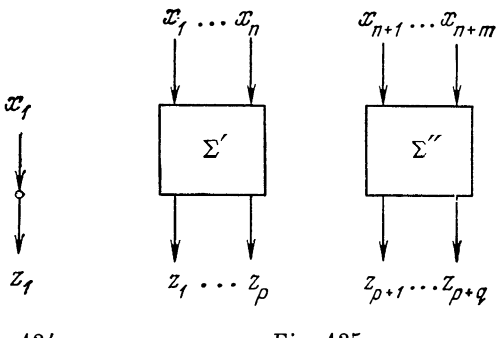


Fig. 124 Fig. 125

tions obtenu à partir de (*) par élimination des lignes $j_1, \ldots, j_{n_i}$ et par adjonction de la ligne

$$z_{p+1} = f_i^0(f_{j_1}(x_1, \ldots, x_n), \ldots, f_{j_{n_i}}(x_1, \ldots, x_n)).$$

On obtient

$$\begin{aligned}
z_1 &= f_1(x_1, \ldots, x_n),\\
&\ldots\ldots\ldots\ldots\\
z_{j_1-1} &= f_{j_1-1}(x_1, \ldots, x_n),\\
z_{j_1+1} &= f_{j_1+1}(x_1, \ldots, x_n),\\
&\ldots\ldots\ldots\ldots\\
z_{j_{n_i}-1} &= f_{j_{n_i}-1}(x_1, \ldots, x_n),\\
z_{j_{n_i}+1} &= f_{j_{n_i}+1}(x_1, \ldots, x_n),\\
&\ldots\ldots\ldots\ldots\\
z_p &= f_p(x_1, \ldots, x_n),\\
z_{p+1} &= f_i^0(f_{j_1}(x_1, \ldots, x_n), \ldots, f_{j_{n_i}}(x_1, \ldots, x_n)).
\end{aligned}$$

III. Supposons que le circuit $\Sigma_3(x_1, \ldots, x_n; z_1, \ldots, z_p, z_{p+1})$ a été obtenu à partir du circuit $\Sigma'(x_1, \ldots, x_n; z_1, \ldots, z_p)$ par

décomposition de la sortie j en deux sorties z_j et z_{p+1}. Associons à ce circuit le système d'équations

$$\begin{aligned} z_1 &= f_1(x_1, \ldots, x_n), \\ &\ldots\ldots\ldots\ldots \\ z_{j-1} &= f_{j-1}(x_1, \ldots, x_n), \\ z_j &= f_j(x_1, \ldots, x_n), \\ z_{j+1} &= f_{j+1}(x_1, \ldots, x_n), \\ &\ldots\ldots\ldots\ldots \\ z_p &= f_p(x_1, \ldots, x_n), \\ z_{p+1} &= f_j(x_1, \ldots, x_n). \end{aligned}$$

Donc, à tout circuit d'éléments fonctionnels on a associé un système d'équations qui en principe est défini par le système de fonctions booléennes figurant dans le second membre.

Exemple 1. a) Pour le circuit Σ_1 (fig. 122) on a le système composé d'une seule équation

$$z_1 = x_1 \,\&\, \overline{x_2} \vee \overline{x_1} \,\&\, x_2 \quad \text{ou} \quad z_1 = x_1 + x_2 \pmod{2}.$$

b) Pour le circuit Σ_2 (fig. 123) on obtient de façon analogue

$$z_1 = x_1 x_4 \vee x_2 x_3.$$

Remarque. Le circuit d'éléments fonctionnels $\Sigma(x_1, \ldots, x_n; z_1, \ldots, z_p)$ peut être traité comme un assemblage des éléments F_i. Chaque élément F_i est un convertisseur élémentaire qui transforme la combinaison d'entrée $(\sigma_1, \ldots, \sigma_{n_i})$ en la valeur de sortie $f_1^0(\sigma_1, \ldots, \sigma_{n_i})$. Dans ce cas, si aux entrées du circuit on applique la combinaison $(\alpha_1, \ldots, \alpha_n)$, alors en se déplaçant des entrées vers les sorties cette combinaison se transformera en la combinaison $(\gamma_1, \ldots, \gamma_p)$ qui caractérise l'état des sorties. Il se trouve que

$$\begin{aligned} \gamma_1 &= f_1(\alpha_1, \ldots, \alpha_n), \\ &\ldots\ldots\ldots\ldots \\ \gamma_p &= f_p(\alpha_1, \ldots, \alpha_n), \end{aligned}$$

c'est-à-dire que la transformation décrite par ces équations correspond à l'idée intuitive que l'on se fait du fonctionnement d'un circuit d'éléments fonctionnels.

Désignons par $\mathfrak{S}$ la classe de tous les circuits d'éléments fonctionnels sur F. Soit $\mathfrak{S}_0$ la sous-classe de tous les circuits d'éléments fonctionnels sur F:

1) possédant exactement une sortie;
2) qui sont ramifiés seulement aux entrées.

Il est évident que $\Sigma_1 \in \mathfrak{S}_0$ et $\Sigma_2 \notin \mathfrak{S}_0$ sur la base F (voir fig. 122, 123). Désignons par $\mathfrak{S}_{\mathfrak{P}}$ la classe des formules sur le système $\mathfrak{P} = \{f_i (x_1, \ldots, x_{n_i})\}$.

Entre les classes $\mathfrak{S}_0$ et $\mathfrak{S}_{\mathfrak{P}}$ il existe une relation qui peut être rigoureusement formulée. En gros, cette relation s'exprime de la manière suivante : à chaque circuit Σ de $\mathfrak{S}_0$ on peut associer de façon unique une formule $\mathfrak{A}$ de $\mathfrak{S}_{\mathfrak{P}}$ de telle sorte que le circuit initial se restaure dans un certain sens de façon unique d'après la formule $\mathfrak{A}$ au symbole près affecté à la sortie.

Cette affirmation peut être illustrée sur un exemple. Considérons le circuit Σ_1 de la figure 122. Il est évident que $\Sigma_1 \in \mathfrak{S}_0$. A ce circuit est associée la formule

$$\mathfrak{A} = (x_1 \& \bar{x}_2) \vee (\bar{x}_1 \& x_2)$$

(voir la construction des équations correspondantes). Il est évident que cette formule restaure la structure du circuit initial Σ.

La démonstration de cette proposition n'est pas compliquée. Il suffit en effet de comparer les définitions d'un circuit de $\mathfrak{S}_0$ et d'une formule sur le système correspondant.

D'après ce qui a été dit, les circuits de $\mathfrak{S}_0$ peuvent être traités comme des formules ou, plus exactement, comme les analogues géométriques des formules.

On remarquera que si un circuit Σ contient un élément F_i présentant une ramification à la sortie, on peut remplacer ce dernier par un groupe d'éléments F_i ne possédant pas de ramifications à la sortie (voir fig. 126) sans pour autant changer le fonctionnement du circuit. Ce remplacement permet de transformer le circuit Σ en un circuit Σ' dont le fonctionnement est régi par les mêmes équations, mais présente des ramifications uniquement aux entrées. Donc, les possibilités fonctionnelles de la classe $\mathfrak{S}$ des circuits Σ sur la base donnée coïncident avec les possibilités fonctionnelles de la sous-classe des circuits Σ' ne présentant des ramifications qu'aux entrées. Chaque circuit Σ' de cette sous-classe est un « raccordement » des circuits de $\mathfrak{S}_0$ par leurs entrées. De là il résulte que les possibilités fonctionnelles de la classe $\mathfrak{S}$ sont définies par celles de la sous-classe $\mathfrak{S}_0$, c'est-à-dire par celles de $\mathfrak{S}_{\mathfrak{P}}$. On vient de démontrer le

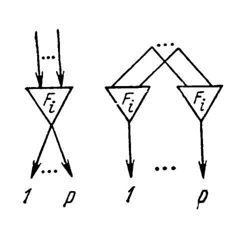


Fig. 126

Théorème 1. *Pour qu'un système arbitraire d'équations booléennes de la forme*

$$z_1 = f_1 (x_1, \ldots, x_n),$$
$$\ldots\ldots\ldots\ldots$$
$$z_p = f_p (x_1, \ldots, x_n),$$

admette dans la base initiale d'éléments fonctionnels un circuit $\Sigma (x_1, \ldots, x_n; z_1, \ldots, z_p)$ *qui le réalise, il est nécessaire et suffisant que le système* $\mathfrak{P}$ *de fonctions* f_i^0 *soit complet.*

On constate donc que la relation existant entre les classes $\mathfrak{S}_0$ et $\mathfrak{S}_{\mathfrak{P}}$ permet de ramener l'étude des possibilités fonctionnelles de la classe $\mathfrak{S}$ à la complétude du système de formules.

On se propose d'étudier maintenant les classes $\mathfrak{S}$ de circuits d'éléments fonctionnels pour lesquels le système $\mathfrak{P}$ est complet.

§ 45. Problème de synthèse de circuits d'éléments fonctionnels

Le problème de synthèse des circuits d'éléments fonctionnels se pose dans les termes suivants. Etant donnés une base F d'éléments fonctionnels et un système arbitraire d'équations booléennes

$$z_1 = f_1 (x_1, \ldots, x_n),$$
$$\ldots\ldots\ldots\ldots$$
$$z_p = f_p (x_1, \ldots, x_n),$$

on demande de construire un circuit $\Sigma (x_1, \ldots, x_n; z_1, \ldots, z_p)$ sur F réalisant ce système d'équations.

Nous avons vu plus haut que le problème de synthèse admet une solution si le système $\mathfrak{P}$ de fonctions f_i^0 est complet. Comme les fonctions booléennes sont réalisées par des formules différentes, le système d'équations booléennes considéré (sous réserve de la complétude de $\mathfrak{P}$) est réalisé par plusieurs circuits d'éléments fonctionnels.

Donc, le problème de synthèse implique des précisions sur le choix d'une solution optimale. Ceci peut être réalisé de la manière suivante. Considérons une fonctionnelle $L(\Sigma)$ définie pour tous les Σ de $\mathfrak{S}$, par exemple comme le nombre des éléments du circuit Σ. Appelons le nombre $L(\Sigma)$ *complexité du circuit.* On exige de plus que le circuit Σ soit de complexité minimale. Dans la suite de tels circuits seront appelés *minimaux.* Donc, le problème de synthèse consistera dorénavant à construire des circuits minimaux.

On se propose dans l'immédiat de décrire un algorithme pour la construction des circuits minimaux. A cet effet, introduisons une notion et prouvons trois lemmes.

Soient donnés n entrées $x_{i_1}, \ldots, x_{i_n}$, p sorties $z_{j_1}, \ldots, z_{j_p}$ et r éléments $F_1, \ldots, F_r$ que nous représenterons comme sur la figure 127.

Définition. On appelle *connexion S* des entrées, sorties et éléments donnés la figure géométrique composée de ces objets et possédant les propriétés suivantes:

1) chaque entrée des éléments est branchée soit à l' entrée, soit à la sortie d'un élément;

2) chaque sortie est branchée soit à l'entrée, soit à la sortie d'un élément.

Remarque. Il est évident que le circuit Σ ($x_{i_1}, \ldots, x_{i_n}$; $z_{j_1}, \ldots, z_{j_p}$) d'éléments fonctionnels est une connexion. Il existe cependant des connexions qui ne sont pas des circuits d'éléments fonctionnels (voir fig. 128).

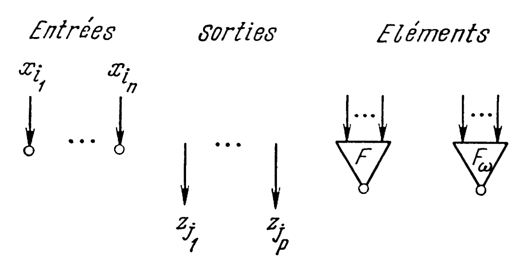


Fig. 127

Lemme 1. *Le nombre $S^*(n, p, h)$ de connexions composées des entrées $x_{i_1}, \ldots, x_{i_n}$, des sorties $z_{j_1}, \ldots, z_{j_p}$ et de h éléments fonctionnels n'est pas supérieur à*

$$H_r^h (n+h)^{h\nu+p},$$

où $\nu = \max_{1 \leqslant i \leqslant r} n_i$.

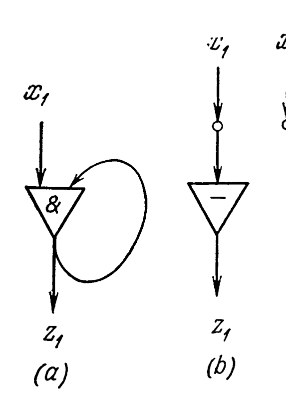


Fig. 128

Démonstration. Choisissons h éléments fonctionnels parmi $F_1, \ldots, F_r$. On peut le faire de H_r^h manières. Branchons les sorties $z_{j_1}, \ldots, z_{j_p}$ et les entrées des éléments (leur nombre est $\leqslant h\nu + p$) soit aux entrées soit aux sorties des éléments (le nombre de possibilités est $n+h$). Donc, pour l'échantillon d'éléments considéré le nombre de branchements n'est pas supérieur à

$$(n+h)^{h\nu+p}.$$

Ceci prouve le lemme.

Corollaire. *Le nombre $S(n, p, h)$ de circuits d'éléments fonctionnels composés des entrées $x_{i_1}, \ldots, x_{i_n}$, des sorties $z_{j_1}, \ldots, z_{j_p}$ et de h éléments fonctionnels vérifie l'inégalité*

$$S(n, p, h) \leqslant S^*(n, p, h) \leqslant H_r^h (n+h)^{h\nu+p} \leqslant r^h (n+h)^{h\nu+p}.$$

Il est immédiat de voir que les opérations définies plus haut pour les circuits logiques peuvent être généralisées aux connexions.

I. Réunion de deux connexions. Cette opération s'applique à deux connexions S' et S'' sans entrées, sorties et éléments communs. La réunion est une connexion possédant tous les éléments et toutes les liaisons de S' et S''. Ses entrées et sorties sont celles de S' et S''.

II. Branchement d'un élément. On considère une connexion S' à p sorties et un élément F_i, $n_i \leqslant p$. Dans S' on choisit n_i sorties (affectées des symboles de l'alphabet z) et on les branche à l'élément F_i ; à la sortie de F_i on affecte un symbole de z différent de ceux affectés aux autres sorties.

III. Ramification d'une sortie. Dans la connexion S' on considère une sortie z_{j_t} branchée soit à la sortie soit à une entrée d'un élément. On prend une nouvelle sortie z (le symbole z est différent des symboles des autres sorties) et on la connecte au même sommet que z_{j_t}.

Lemme 2. *Les opérations* I, II *et* III *appliquées à des connexions donnent une connexion.*

Lemme 3. *Une connexion différente d'un circuit trivial est un circuit si et seulement si est réalisée l'une au moins des conditions suivantes*:

1) *S est la réunion des connexions S' et S'' qui sont des circuits*;

2) *S se déduit à partir de S' par branchement d'un élément F_i et S' est un circuit*;

3) *S se déduit à partir de S' par ramification de sa sortie et S' est un circuit.*

La démonstration de ces deux lemmes est évidente.

Théorème 2. *Il existe un algorithme disant si S est circuit ou non.*

Démonstration. Elle se fait par récurrence sur le nombre d de liaisons de la connexion S, c'est-à-dire sur le nombre total des entrées des éléments et des sorties de S. Lorsque $d = 1$, alors ou bien S coïncide avec un circuit trivial ou bien n'est pas un circuit. Supposons que le théorème est vrai pour tous les $d = 1, \ldots, \lambda$. Montrons qu'il l'est pour $d = \lambda + 1$. Considérons une connexion S à $\lambda + 1$ liaisons. Il est évident que S est différente d'un circuit trivial ($\lambda \geqslant 2$). Deux cas sont possibles pour la connexion S:

1) S ne se déduit à partir de connexions présentant un nombre inférieur de liaisons par l'application d'aucune des opérations I, II et III. Il est alors évident que S n'est pas un circuit ;

2) S s'obtient à partir de connexions présentant un nombre inférieur de liaisons.

Dans ce cas on considère toutes les décompositions possibles (celles-ci sont en nombre fini) et en s'appuyant sur l'hypothèse de la récurrence concernant les composantes de la décomposition on voit si ces composantes sont des circuits ou non. S'il existe une décomposition dont toutes les composantes sont des circuits, alors la connexion initiale sera un circuit en vertu du lemme précédent. Si dans chaque décomposition l'une au moins des composantes n'est un circuit, alors la connexion ne le sera pas non plus. C.q.f.d.

Théorème 3. *Il existe un algorithme qui pour chaque système d'équations booléennes construit un circuit minimal* Σ.

Démonstration. Soit le système d'équations

$$z_1 = f_1 (x_1, \ldots, x_n),$$
$$\ldots\ldots\ldots\ldots$$
$$z_p = f_p (x_1, \ldots, x_n).$$

Etudions successivement toutes les connexions à n entrées $x_1, \ldots, x_n$, p sorties $z_1, \ldots, z_p$ et $h = 0, 1, 2, \ldots$ éléments.

Soit $h = 0$. Nous avons un nombre fini de connexions ne contenant pas d'éléments. Vérifions si chaque connexion est un circuit ou non. Si la connexion considérée est un circuit, voyons si elle réalise le système d'équations donné. Donc, ou bien nous trouvons le circuit cherché (et celui-ci sera manifestement minimal), ou bien le circuit cherché n'est pas un circuit de complexité $h = 0$. Auquel cas le circuit minimal sera de complexité $L(\Sigma) > 0$.

Soit $h = 1$. Nous avons un nombre fini de connexions contenant un élément. Vérifions si chaque connexion est un circuit ou non. Si oui, voyons si elle réalise le système d'équations donné. Donc, soit on trouve le circuit cherché, et celui-ci sera minimal, soit le circuit cherché n'est pas de complexité $h = 0$ et $h = 1$. Par suite, le circuit minimal Σ sera de complexité $L(\Sigma) > 1$, et ainsi de suite. Le système $\mathfrak{P}$ étant complet, ce processus prend fin et nous obtenons un circuit minimal. C.q.f.d.

Cet algorithme de construction de circuits minimaux se rapporte à la classe des algorithmes de type « examen complet » car il repose sur l'examen de tous les circuits jusqu'à une certaine complexité. Ces algorithmes sont en principe très laborieux. Ils ne permettent pas en général d'étudier les propriétés des objets cherchés et sont impropres à tout usage pratique. Pour cette raison on étudie plus bas un problème plus simple dans lequel le système d'équations n'est composé que d'une seule équation

$$z = f(x_1, \ldots, x_n),$$

par suite le circuit cherché Σ sera à une seule sortie.

Désignons par $L(f)$ la complexité du circuit minimal.

De plus, au lieu de la synthèse des circuits pour des fonctions (équations), on étudie le problème de la synthèse pour la classe $P^{(n)}$ de toutes les fonctions de n variables. Ceci étant, on teste la qualité des algorithmes de synthèse par comparaison des fonctions de Shannon. De façon plus exacte, soit

$$L(n) = \max_{f \in P^{(n)}} L(f), \quad L_A(n) = \max_{f \in P^{(n)}} L_A(f),$$

où $L_A(f)$ est la complexité minimale des circuits réalisant f, obtenus à l'aide de l'algorithme A.

Les fonctions $L(n)$ et $L_A(n)$ s'appellent *fonctions de Shannon.* Elles caractérisent de toute évidence la complexité de la classe $P^{(n)}$ du point de vue de la réalisation de ses fonctions par des circuits minimaux (respectivement par des circuits minimaux relativement à l'algorithme A) et

$$L_A(n) \geqslant L(n).$$

Le problème de synthèse consiste maintenant à trouver un algorithme A tel que $L_A(n)$ soit le plus proche de $L(n)$ (par exemple, $L_A(n) = L(n)$) et que sa taille soit sensiblement moindre que celle de l'algorithme d'examen complet. A noter que dans la nouvelle position du problème on n'exige plus que l'algorithme A trouve un circuit minimal pour chaque fonction f, on demande simplement que le circuit le plus élémentaire obtenu à l'aide de A possède une complexité $L_A(f)$ qui ne soit pas de beaucoup supérieure à $L(n)$.

§ 46. Méthodes élémentaires de synthèse

Nous allons citer quelques algorithmes de synthèse reposant sur des idées élémentaires dans le cas où la base F est constituée d'un inverseur, d'un circuit « $\bigvee$ » et d'un circuit « & ».

a) *Méthode de synthèse basée sur la forme canonique disjonctive.*

Considérons la représentation de la fonction $f(x_1, \ldots, x_n)$ ($f \not\equiv$ const) par la forme canonique disjonctive:

$$f(x_1, \ldots, x_n) = \bigvee_{\substack{(\sigma_1 \ldots \sigma_n) \\ f(\sigma_1, \ldots, \sigma_n)=1}} x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n} = \bigvee_{i=1}^{s} K_i.$$

Introduisons un «élément» auxiliaire (fig. 129) permettant de représenter facilement (fig. 130) le circuit Σ_K réalisant la conjonction

$$K = x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}.$$

Il est évident que $L(\Sigma_K) \leqslant n + (n-1)$ et que Σ_K contient un sous-circuit Σ'_K qui est le même pour toutes les conjonctions et dont la complexité est $n-1$. Si l'on « raccorde » les circuits $\Sigma_{K_1}, \ldots$

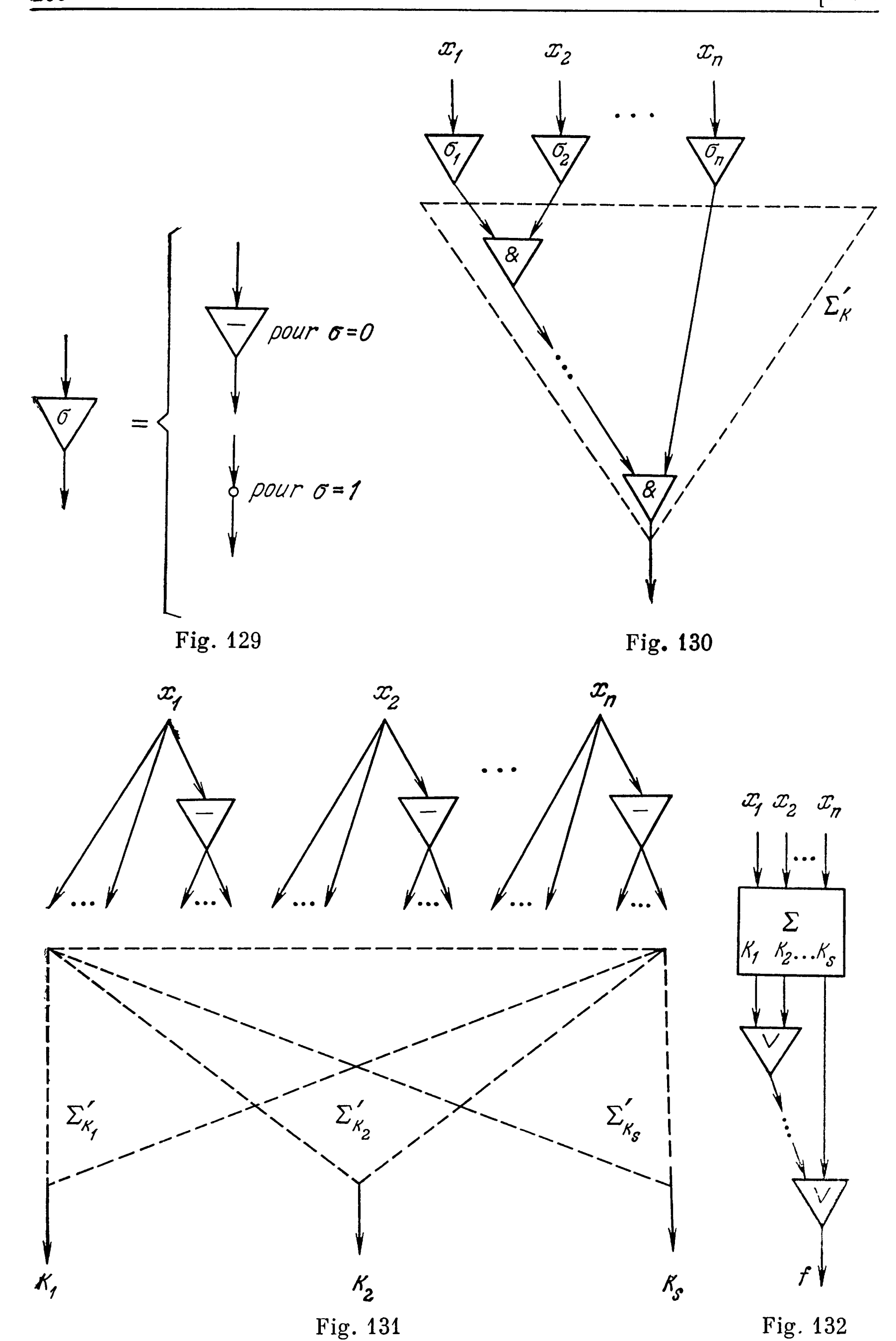


Fig. 129

Fig. 130

Fig. 131

Fig. 132

. . ., Σ_{K_s} en commençant par les entrées $x_1, \ldots, x_n$ et jusqu'aux éléments auxiliaires, on obtient le circuit Σ (fig. 131).

On a $L(\Sigma) \leqslant n + s(n-1)$. En connectant les sorties du circuit Σ aux circuits « $\bigvee$ », on obtient un circuit pour $f(x_1, \ldots, x_n)$ (fig. 132).

Ceci achève la description de la méthode de synthèse pour la forme canonique disjonctive (algorithme A_1). On obtient en définitive

$$L_{A_1}(f) \leqslant n + s(n-1) + s - 1 < n + ns = n(s+1).$$

Comme $f \not\equiv 1$, il vient que $s \leqslant 2^n - 1$ et

$$L_{A_1}(f) \leqslant n2^n$$
$$L_{A_1}(n) \leqslant n2^n.$$

b) *Méthode de synthèse basée sur une réalisation plus compacte de l'ensemble de toutes les conjonctions*

$$\{x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}\}.$$

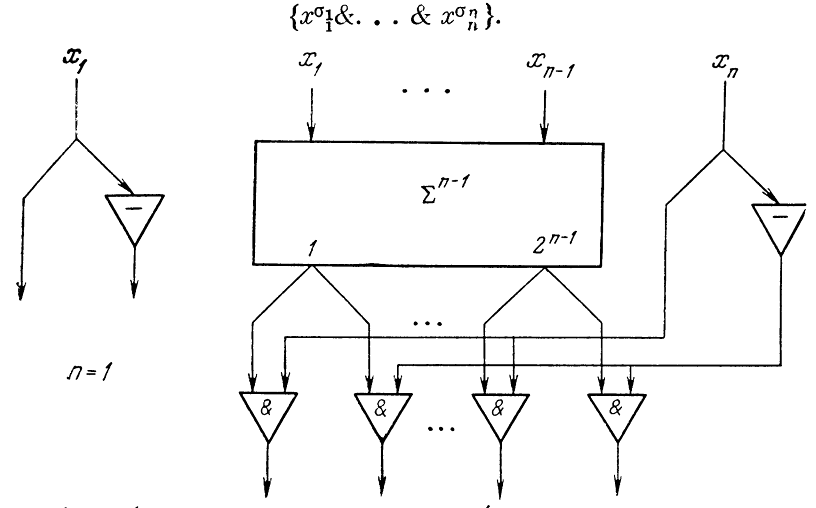


Fig. 133

La figure 133 représente la construction par récurrence d'un multipôle Σ^n $(n = 1, 2, \ldots)$ réalisant l'ensemble de toutes les conjonctions $\{x_1^{\sigma_1} \& \ldots \& x_n^{\sigma_n}\}$. On a

$$L(\Sigma^1) = 1,$$
$$L(\Sigma^n) = L(\Sigma^{n-1}) + 1 + 2^n,$$
$$L(\Sigma^n) = 2^2 + \ldots + 2^n + n = 2 \cdot 2^n + n - 4.$$

Pour construire le circuit qui réalise la fonction $f(x_1, \ldots, x_n)$ il faut dans le multipôle Σ^n sélectionner les sorties correspondant aux termes $K_1, \ldots, K_s$ de sa forme canonique disjonctive, les connecter au circuit (cf. fig. 132) réalisant l'addition logique et supprimer les éléments superflus. Ceci implique encore

$$s \leqslant 2^n - 1$$

éléments $\vee$.

Donc, cette méthode (algorithme A_2) donne

$$L_{A_2}(f) \leqslant 3 \cdot 2^n + n - 5 \text{ et } L_{A_2}(n) \leqslant 3 \cdot 2^n + n - 5.$$

c) *Méthode de synthèse basée sur le développement de la fonction $f(x_1, \ldots, x_n)$ suivant la variable x_n.*

Considérons le développement

$$f(x_1, \ldots, x_{n-1}, x_n) = x_n \,\&\, f(x_1, \ldots, x_{n-1}, 1) \vee \bar{x}_n \,\&\, f(x_1, \ldots, x_{n-1}, 0)$$

et pour simplifier posons

$$f' = f(x_1, \ldots, x_{n-1}, 1),\ f'' = f(x_1, \ldots, x_{n-1}, 0).$$

La figure 134 représente la procédure de construction par récurrence du circuit pour f.

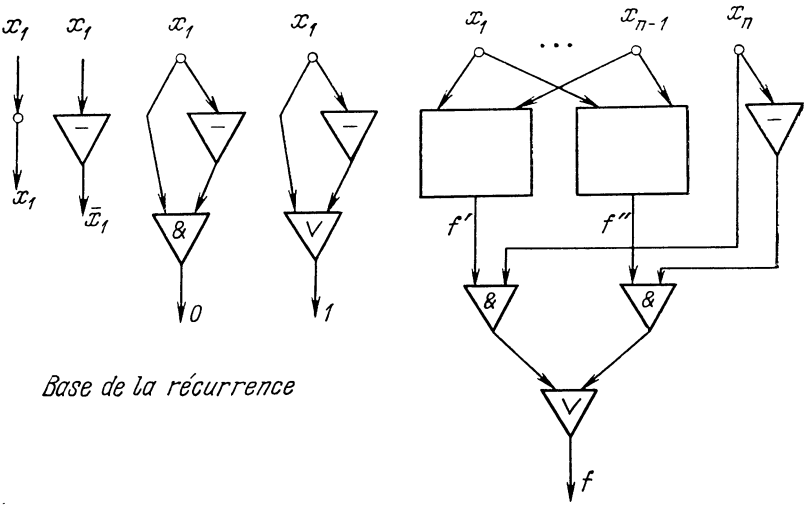


Fig. 134

D'après cette méthode on a l'algorithme A_3:

$$L_{A_3}(1) = 2,$$
$$L_{A_3}(n) \leqslant 2L_{A_3}(n-1) + 4.$$

On obtient en définitive

$$L_{A_3}(n) \leqslant 3 \cdot 2^n - 4.$$

Si l'on tient compte de l'information sur la réalisation des fonctions de deux variables, par exemple du fait que $L_{A_3}(2) \leqslant 5$, on peut obtenir une meilleure majoration, soit

$$L_{A_3}(n) \leqslant 2{,}25 \cdot 2^n - 4.$$

On voit que les algorithmes A_1, A_2 et A_3 permettent dans un certain sens d'obtenir des réalisations plus compactes pour les fonctions et en fin de compte de meilleures majorations pour les fonctions de Shannon. D'autre part, on obtient de meilleurs résultats de synthèse en compliquant légèrement l'algorithme.

§ 47. Minoration de $L(n)$

Pour juger de la qualité des algorithmes de synthèse il faut mesurer l'écart entre $L_A(f)$ et $L(f)$. Cette comparaison est impossible à faire, car le calcul de la quantité $L_A(f)$ est assez compliqué (par exemple, pour les algorithmes A_2 et A_3) et la quantité $L(f)$ est inconnue.

Pour cette raison, la qualité d'un algorithme de synthèse s'estime par comparaison des fonctions de Shannon $L_A(n)$ et $L(n)$. Malheureusement, nous ne pouvons comparer $L_A(n)$ et $L(n)$ pour chaque valeur de n, car si le calcul de $L_A(n)$ est plus aisé que celui de $L_A(f)$, le calcul de $L(n)$ est très compliqué. Nous sommes donc contraints de nous contenter d'une comparaison asymptotique, c'est-à-dire de comparer $L_A(n)$ et $L(n)$ pour $n \to \infty$.

Définition. Soient $\varphi(n)$ et $\psi(n)$ des fonctions positives réelles d'une variable entière n.

1. *La fonction* $\varphi(n)$ *est asymptotiquement plus grande ou égale à* $\psi(n)$, ce qu'on note $\varphi(n) \gtrsim \psi(n)$, si pour tout $\varepsilon > 0$ il existe $N = N(\varepsilon)$ tel que pour tout $n \geqslant N$ on a $\varphi(n) \geqslant (1-\varepsilon)\psi(n)$.

2. Si $\varphi(n) \gtrsim \psi(n)$ et $\psi(n) \gtrsim \varphi(n)$, on dit que *les fonctions* $\varphi(n)$ *et* $\psi(n)$ *sont asymptotiquement égales* (*équivalentes*) et l'on note $\varphi(n) \sim \psi(n)$. Il est évident que la limite de $\varphi(n)/\psi(n)$ existe et est égale à 1.

3. Si $0 < c_1 < \varphi(n)/\psi(n) < c_2$, on dit que *les fonctions* $\varphi(n)$ *et* $\psi(n)$ *sont équivalentes à l'ordre près*. On écrit alors $\varphi(n) \asymp \psi(n)$.

Pour comparer $L_A(n)$ et $L(n)$ pour $n \to \infty$ il importe d'obtenir une minoration asymptotique assez bonne de $L(n)$ pour une base constituée d'un inverseur, d'un élément « & » et d'un élément « $\vee$ ».

Théorème 4. $L(n) \gtrsim \frac{2^{n-1}}{n}$.

Démonstration. On remarquera d'abord que si $f(x_1, \ldots, x_n)$ est réalisable par un circuit Σ' d'éléments fonctionnels de complexité $h' \leqslant h$, alors elle peut manifestement être réalisée par un circuit Σ d'éléments fonctionnels de complexité h. Le circuit Σ se déduit à partir du circuit Σ' comme l'indique la figure 135.

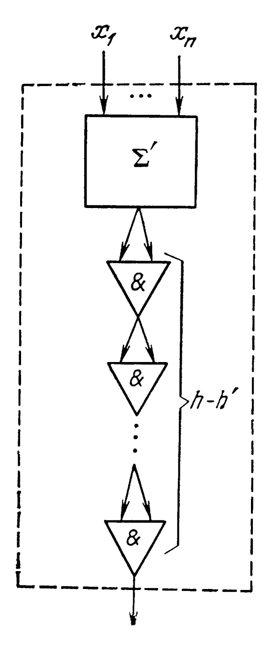


Fig. 135

Nous avons vu plus haut (voir corollaire du lemme 1) que le nombre de circuits d'éléments fonctionnels à n entrées et une sortie ($p = 1$) contenant exactement h éléments dans la base considérée ($\nu = 2$, $r = 3$), c'est-à-dire $S(n, 1, h)$, se majore comme suit:

$$S(n, 1, h) \leqslant 3^h (n + h)^{2h+1}.$$

Si $h > n$, alors pour $h \geqslant N$

$$S(n, 1, h) \leqslant 3^h (n+h)^{2h+1} < (ch)^{2h}.$$

Supposons que $h = \left[(1-\varepsilon)\frac{2^{n-1}}{n}\right]$ et majorons le nombre de circuits d'éléments fonctionnels de complexité h. On a

$$\left(c(1-\varepsilon)\frac{2^{n-1}}{n}\right)^{2(1-\varepsilon)2^{n-1}/n} = \left(c(1-\varepsilon)\frac{2^{n-1}}{n}\right)^{(1-\varepsilon)2^n/n},$$

donc les circuits de complexité h peuvent réaliser au plus

$$\left(c(1-\varepsilon)\frac{2^{n-1}}{n}\right)^{(1-\varepsilon)2^n/n}$$

fonctions booléennes de n variables.

Considérons l'expression

$$\log_2 \frac{\left(c(1-\varepsilon)\frac{2^{n-1}}{n}\right)^{(1-\varepsilon)2^n/n}}{2^{2^n}} = (1-\varepsilon)\frac{2^n}{n}\left((n-1) + \log\frac{c(1-\varepsilon)}{n}\right) - 2^n =$$

$$= \frac{2^n}{n}\left((1-\varepsilon)\log\frac{c(1-\varepsilon)}{n} - \varepsilon n - 1 + \varepsilon\right) \to -\infty \quad (n \to \infty).$$

Donc, pour $n > N_0$ le numérateur sera inférieur au dénominateur. Par suite, les circuits de complexité $\leqslant h$ ne suffisent pas pour réaliser toutes les fonctions booléennes de n variables. Il existe par consé-

quent des fonctions de n variables qui ne peuvent être réalisées avec une complexité inférieure ou égale à

$$h=\left[(1-\varepsilon)\frac{2^{n-1}}{n}\right],$$

c'est-à-dire

$$L(n)>(1-\varepsilon)\frac{2^{n-1}}{n} \quad \text{ou} \quad L(n)\gtrsim\frac{2^{n-1}}{n}.$$

C.q.f.d.

Corollaire. *Le nombre de fonctions f telles que $L(f)>(1-\varepsilon)\frac{2^{n-1}}{n}$ tend vers 1 pour $n\to\infty$.*

§ 48. Méthode de synthèse de circuits d'éléments fonctionnels d'ordre minimal (méthode de Shannon)

La minoration obtenue plus haut pour $L(n)$ montre que les meilleures méthodes élémentaires de synthèse ont un ordre n fois supérieur. Cela veut dire que l'amélioration des méthodes de synthèse peut entraîner une réduction de l'ordre de la fonction de Shannon de n fois au plus par rapport à $c2^n$. En fait, il se trouve qu'il existe une méthode de synthèse qui conduit à une fonction de Shannon dont l'ordre est confondu avec la minoration. Cette méthode a été élaborée par Shannon pour les circuits à contacts. Nous allons l'exposer pour la réalisation de fonctions booléennes par des circuits d'éléments fonctionnels.

Soit $\{f_1(x_1, \ldots, x_n), \ldots, f_s(x_1, \ldots, x_n)\}$ un ensemble de fonctions booléennes ($f_i \neq f_j$ pour $i\neq j$).

Définition. On dit d'un multipôle d'éléments fonctionnels à n entrées et s sorties qu'il est *universel* pour un ensemble donné de fonctions si pour tout i ($1\leqslant i\leqslant s$) il présente une sortie $\tau(i)$ qui délivre une fonction $f_i(x_1, \ldots, x_n)$ de cet ensemble.

Exemple 2. Soit $\{K_1, \ldots, K_s\}$ un ensemble de conjonctions. Alors le multipôle (cf. fig. 131) est universel pour cet ensemble.

Lemme 4. *Pour tout n on peut construire un multipôle universel U_n pour l'ensemble de toutes les fonctions booléennes de n variables $x_1, \ldots, x_n$ et*

$$L(U_n)\leqslant 2\cdot 2^{2^n}.$$

Démonstration. Nous allons construire le multipôle U_n par récurrence.

B a s e   d e   l a   r é c u r r e n c e ($n=1$). Pour U_1 prenons le multipôle représenté sur la figure 136. On a $L(U_1)=3<2\cdot 2^{2^1}$.

P a s s a g e d e r é c u r r e n c e. Supposons qu'on a construit un multipôle universel U_{n-1} pour l'ensemble de toutes les fonctions booléennes dépendant des variables $x_1, \ldots, x_{n-1}$, et que $L(U_{n-1}) \leqslant 2 \cdot 2^{2^{n-1}}$. Considérons le développement

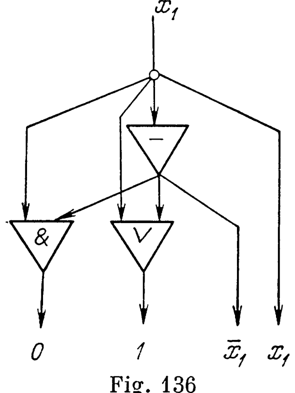


Fig. 136

$$f(x_1, \ldots, x_{n-1}, x_n) = x_n f'(x_1, \ldots$$
$$\ldots, x_{n-1}) \vee \bar{x}_n f''(x_1, \ldots, x_{n-1}).$$

Partageons l'ensemble de toutes les fonctions booléennes dépendant des variables $x_1, \ldots, x_n$ en trois classes disjointes.

I. $f' \equiv f'' \equiv 0$. Cette classe contient une seule fonction $f \equiv 0$.

II. Exactement une des fonctions f' ou f'' est identiquement égale à 0. Cette classe contient les fonctions f de la forme

$$f = x_n f'(x_1, \ldots, x_{n-1}) \quad (f' \not\equiv 0)$$

ou

$$f = \bar{x}_n f''(x_1, \ldots, x_{n-1}) \quad (f'' \not\equiv 0),$$

c'est-à-dire $2(2^{2^{n-1}} - 1)$ fonctions.

III. Toutes les fonctions restantes, c'est-à-dire les fonctions f telles que

$$f' \not\equiv 0 \quad \text{et} \quad f'' \not\equiv 0.$$

Cette classe contient

$$2^{2^n} - 2(2^{2^{n-1}} - 1) - 1$$

fonctions.

La figure 137 représente un multipôle U_n. Les sorties des multipôles U_{n-1} sont numérotées par les nombres $1, \ldots, 2^{2^{n-1}}$ et on admet de plus que la constante 0 est réalisée à la sortie 1.

Les sorties du multipôle U_n sont réparties en trois classes selon la partition de l'ensemble de toutes les fonctions booléennes dépendant des variables $x_1, \ldots, x_n$. Le multipôle U_n contient:

a) un sous-circuit U_{n-1} et en dehors de ce sous-circuit encore,

b) un inverseur,

c) $2(2^{2^{n-1}} - 1)$ éléments « & »,

d) $2^{2^n} - 2(2^{2^{n-1}} - 1) - 1$ éléments « $\vee$ ».

Donc

$$L(U_n) = L(U_{n-1}) + 1 + 2(2^{2^{n-1}} - 1) + 2^{2^n} - 2(2^{2^{n-1}} - 1) - 1 =$$
$$= L(U_{n-1}) + 2^{2^n} \leqslant 2 \cdot 2^{2^{n-1}} + 2^{2^n} \leqslant 2 \cdot 2^{2^n}.$$

Ceci achève la démonstration du lemme.

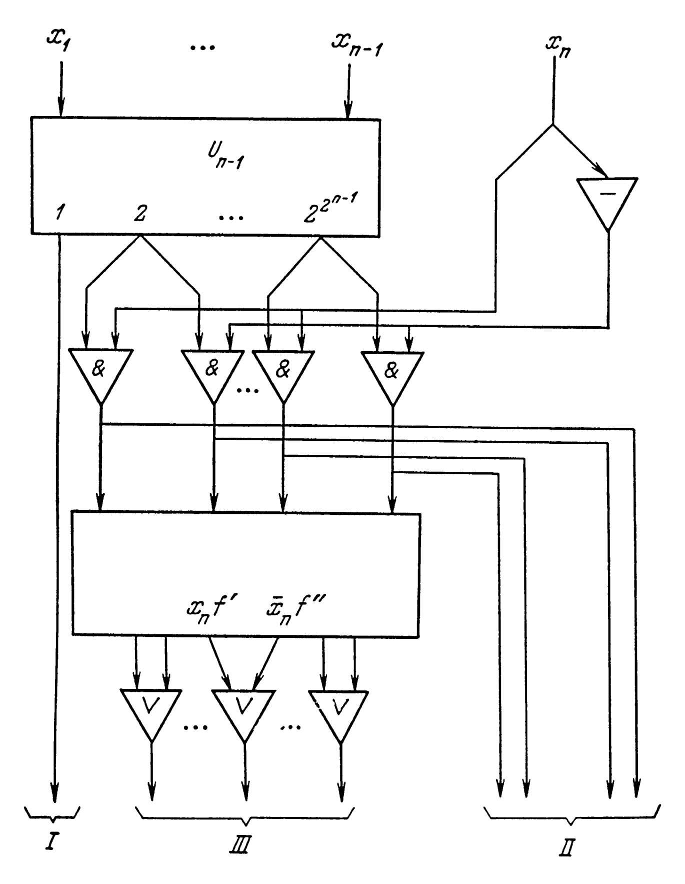


Fig. 137

Théorème 5. *Il existe une méthode de synthèse (algorithme A_4) qui permet pour chaque fonction booléenne $f(x_1, \ldots, x_n)$ de construire un circuit d'éléments fonctionnels de complexité $L_{A_4}(f)$ et*

$$L_{A_4}(f(x_1, \ldots, x_n)) \leqslant 8\frac{2^n}{n}.$$

Démonstration. Soit la décomposition

$$f(x_1, \ldots, x_n) = \bigvee_{(\sigma_1, \ldots, \sigma_k)} x_1^{\sigma_1} \& \ldots \& x_k^{\sigma_k} \& f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n).$$

Considérons le circuit Σ_f de la figure 138.

V_k est un multipôle universel pour l'ensemble de toutes les conjonctions

$$K_{\sigma_1 \ldots \sigma_k} = x_1^{\sigma_1} \& \ldots \& x_k^{\sigma_k}$$

(k est un nombre fixe). Pour V_k on prend le circuit Σ^k (cf. fig. 133).

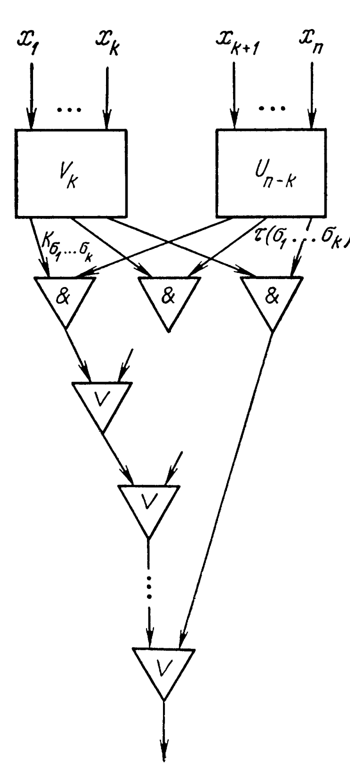


Fig. 138

U_{n-k} est un multipôle universel pour l'ensemble de toutes les fonctions booléennes des variables $x_{k+1}, \ldots, x_n$. Par $\tau(\sigma_1, \ldots, \sigma_k)$ on désigne la sortie qui réalise la fonction

$$f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n).$$

Il est immédiat de voir que le circuit Σ_f réalise la fonction $f(x_1, \ldots, x_n)$ d'après la décomposition mentionnée. Majorons la complexité de Σ_f:

$$L(\Sigma_f) \leqslant L(V_k) + L(U_{n-k}) + 2 \cdot 2^k - 1 \leqslant \leqslant 2 \cdot 2^k + k - 4 + 2 \cdot 2^{2^{n-k}} + + 2 \cdot 2^k - 1 = 4 \cdot 2^k + 2 \cdot 2^{2^{n-k}} + k - 5.$$

Choisissons la valeur du paramètre k qui minimise le second membre de cette inégalité. Comme nous nous intéressons à l'ordre de la quantité $L_{A_4}(f)$, au lieu du minimum du second membre dont le calcul est compliqué dans le cas discret, nous allons prendre la valeur du second membre pour $m = [\log_2 (n - 2 \log_2 n)]$, où $m = n - k$. On choisit cette valeur à partir des considérations suivantes: lorsque k croît, le terme $4 \cdot 2^k$ croît et $2 \cdot 2^{2^{n-k}}$ décroît, et le minimum est réalisé quand ces deux termes sont assez voisins l'un de l'autre.

On a

$$\frac{1}{2}(n-2\log n) < 2^m < (n-2\log n),$$

$$L_{A_4}(f) \leqslant 4\frac{2^n}{2^m} + 2\cdot 2^{2^m} \leqslant \frac{4\cdot 2^n}{\frac{1}{2}(n-2\log n)} + 2\frac{2^n}{n^2} \leqslant 8\frac{2^n}{n}.$$

Corollaire. *En tenant compte de la minoration on obtient*

$$L(n) \asymp \frac{2^n}{n},$$

donc l'algorithme A_4 est le meilleur pour l'ordre.

Des méthodes de synthèse plus fines permettent d'obtenir le résultat suivant (nous omettons la démonstration).

Théorème 6 (O. Loupanov [10]). *Pour les circuits d'éléments fonctionnels on peut construire dans une base composée d'un inverseur d'un élément « $\vee$ » et d'un élément « & » la méthode de synthèse la meilleure asymptotiquement et*

$$L(n) \sim \frac{2^n}{n}.$$

§ 49. Synthèse d'un sommateur

La théorie générale de la synthèse des circuits d'éléments fonctionnels nous conduit à une conclusion importante, savoir que la plupart des fonctions booléennes (pour $n \to \infty$) possèdent des circuits minimaux compliqués. Cela veut dire que seule une classe assez étroite de fonctions booléennes présente un intérêt pratique du point de vue de la synthèse. Donc, outre les méthodes universelles de synthèse il faut se munir de méthodes de synthèse adaptées à certaines classes de fonctions booléennes et tirant le meilleur parti des propriétés de ces fonctions.

Dans ce paragraphe, nous allons étudier une approche de réalisation d'une classe assez étroite de fonctions. Il sera question de la construction d'un circuit multipôle d'éléments fonctionnels réalisant l'addition de deux nombres binaires

$$x = x_n x_{n-1} \ldots x_1$$

et

$$y = y_n y_{n-1} \ldots y_1.$$

Considérons à cet effet l'algorithme classique d'addition des nombres x et y rang par rang

$$\begin{array}{r} + \begin{array}{r} q_{n+1}q_n \ldots q_1 \\ x_n \ldots x_1 \\ y_n \ldots y_1 \end{array} \\ \hline z_{n+1}z_n \ldots z_1 \end{array}$$

Les nombres $q_{n+1}, \ldots, q_1$ représentent les reports des rangs précédents ($q_1 = 0$). Il est évident que

$$z_i = x_i + y_i + q_i \pmod{2};$$

$$q_{i+1} = x_iy_i \vee x_iq_i \vee y_iq_i.$$

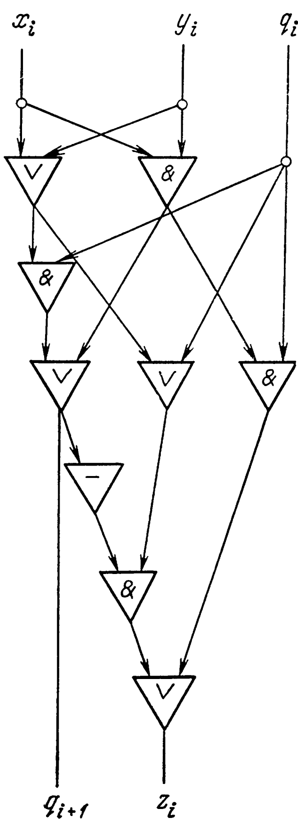


Fig. 139

En se servant de l'identité

$$x_i + y_i + q_i = \\ = \overline{(x_iy_i \vee x_iq_i \vee y_iq_i)} \mathbin{\&} (x_i \vee \\ \vee y_i \vee q_i) \vee x_iy_iq_i,$$

on obtient sans peine le circuit qui réalise la conversion correspondante des quantités x_i, y_i, q_i, en z_i, q_{i+1} (cf. fig. 139). Désignons ce circuit par B_i ($1 < i \leqslant n$). Le circuit cherché Σ_n qui est un sommateur pour deux nombres binaires à n positions s'obtient par connexion en série des blocs B_i (cf. fig. 140). Ici $z_{n+1} = q_{n+1}$ et le bloc B_1 réalise la conversion

$$z_1 = x_1 + y_1 = \overline{x_1y_1}\,(x_1 \vee y_1)$$
$$q_2 = x_1y_1.$$

Il est évident que $L(B_1) = 4$ et $L(B_i) = 9$ pour $1 < i \leqslant n$. Donc

$$L(\Sigma_n) \leqslant 9(n-1) + \\ + 4 = 9n - 5 < 9n.$$

§ 50. Synthèse de circuits d'éléments fonctionnels réalisant des fonctions symétriques

Une autre classe intéressante de conversions binaires est la classe des fonctions symétriques.

Définition. On dit qu'une fonction booléenne $S(x_1, \ldots, x_n)$ est *symétrique* si pour toute combinaison $(\alpha_1, \ldots, \alpha_n)$ et toute permutation $\begin{pmatrix} 1 \ldots n \\ j_1 \ldots j_n \end{pmatrix}$ on a

$$S(\alpha_{j_1}, \ldots, \alpha_{j_n}) = S(\alpha_1, \ldots, \alpha_n).$$

La classe des fonctions symétriques contient visiblement les constantes 0 et 1, les fonctions $\bar{x}_1$, $x_1 \& x_2$, $x_1 \vee x_2$, $x_1 + x_2$, $x_1 x_2 \vee \vee x_1 x_3 \vee x_2 x_3$, etc.

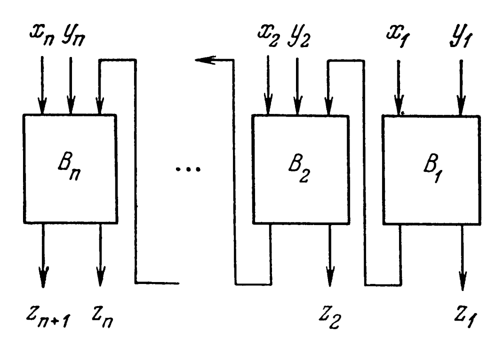


Fig. 140

Il est aisé de voir que si pour une combinaison $(\alpha_1, \ldots, \alpha_n)$ on a

$$S(\alpha_1, \ldots, \alpha_n) = 1$$

et $(\alpha_1, \ldots, \alpha_n)$ contient exactement i unités, alors pour toute combinaison $(\alpha_1', \ldots, \alpha_n')$ contenant aussi exactement i unités on aura

$$S(\alpha_1', \ldots, \alpha_n') = 1.$$

Donc, la fonction symétrique $S(x_1, \ldots, x_n)$ est caractérisée par la liste des nombres de travail $i_1, \ldots, i_r$ $(0 \leqslant i_1 < \ldots < i_r \leqslant n)$ qui représentent le nombre d'unités dans les combinaisons $(\alpha_1, \ldots \ldots, \alpha_n)$ pour lesquelles

$$S(\alpha_1, \ldots, \alpha_n) = 1.$$

On peut donc noter que

$$S(x_1, \ldots, x_n) = S_{i_1, \ldots, i_r}(x_1, \ldots, x_n).$$

Par exemple

$$x_1 x_2 \vee x_1 x_3 \vee x_2 x_3 = S_{2,3}(x_1, x_2, x_3).$$

Considérons la conversion auxiliaire

$$(\alpha_1, \ldots, \alpha_n) \to i_{(\alpha_1, \ldots, \alpha_n)},$$

où $i_{(\alpha_1, \ldots, \alpha_n)}$ est le nombre d'unités dans la combinaison $(\alpha_1, \ldots, \alpha_n)$. Cette conversion sera réalisée par le multipôle Σ_n^1 d'éléments fonctionnels (cf. fig. 141) dont les sorties

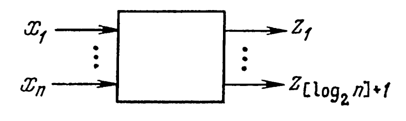


Fig. 141

$z_1, \ldots, z_{[\log_2 n]+1}$ délivreront la représentation binaire du nombre $i_{(\alpha_1, \ldots, \alpha_n)}$ si à l'entrée est appliquée la combinaison $(\alpha_1, \ldots, \alpha_n)$.

Soit $n = 2^m$. Alors $[\log_2 n] + 1 = m + 1$.

Lemme 5. *On peut construire un multipôle Σ'_n réalisant au sens indiqué plus haut la conversion*

$$(\alpha_1, \ldots, \alpha_n) \to i_{(\alpha_1, \ldots, \alpha_n)}$$

et $L(\Sigma'_n) \leqslant 18n - 9\log n - 18$.

Démonstration. Elle se fait par récurrence sur m.

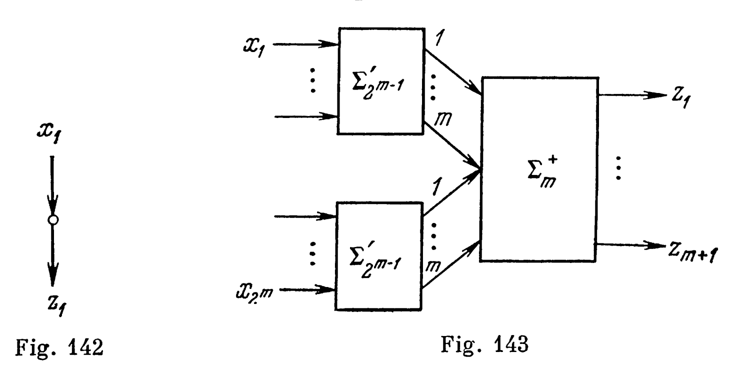


Fig. 142 Fig. 143

a) *Base de la récurrence*: $m = 0$. Ici $n = 1$ et le circuit cherché Σ'_1 est de la forme $L(\Sigma'_1) = 0$ (fig. 142). D'autre part $18n - 9\log_2 n - 18 = 0$ pour $n = 1$. Donc, l'inégalité

$$L(\Sigma'_n) \leqslant 18n - 9\log n - 18$$

est valable pour $n = 1$.

b) *Le passage de la récurrence de* $m-1$ *à* $m\,(m-1\geqslant 0)$ se réalise à l'aide du circuit Σ'_{2^m} (fig. 143). Ce circuit « divise » la combinaison $(\alpha_1, \ldots, \alpha_{2^m})$ en deux parties: $(\alpha_1, \ldots, \alpha_{2^{m-1}})$ et $(\alpha_{2^{m-1}+1}, \ldots, \alpha_{2^m})$. Dans chaque partie les sous-circuits $\Sigma'_{2^{m-1}}$ calculent le nombre i' et i'' d'unités, ensuite le sommateur Σ^+_m

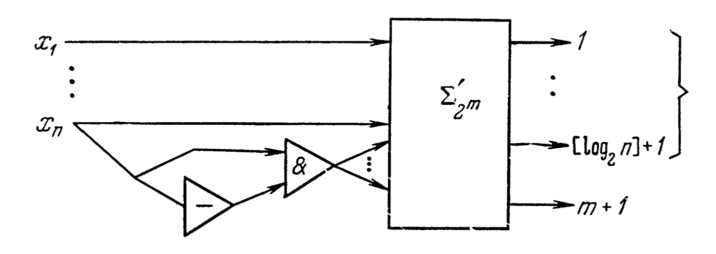


Fig. 144

« additionne » ces nombres et délivre le nombre d'unités de la combinaison initiale.

On a $L(\Sigma'_{2^m}) = 2L(\Sigma'_{2^{m-1}}) + L(\Sigma^+_m)$, ou

$$L(\Sigma'_n) = 2L(\Sigma'_{n/2}) + L(\Sigma^+_m) \leqslant 2\left(18\frac{n}{2} - 9\log\frac{n}{2} - 18\right) + \\ + 9\log_2 n < 18n - 9\log n - 18.$$

Ce qui prouve le lemme.

Corollaire. $L(\Sigma'_{2^m}) \leqslant 18\cdot 2^m$, *c'est-à-dire que pour* $n = 2^m$

$$L(\Sigma'_n) \leqslant 18n.$$

Lemme 6. *Pour tout* n *on peut construire un multipôle* Σ'_n *réalisant la conversion* $(\alpha_1, \ldots, \alpha_n) \to i_{(\alpha_1, \ldots, \alpha_n)}$, *et* $L(\Sigma'_n) \leqslant 36n$.

Démonstration. Il est évident qu'il existe un m tel que

$$n \leqslant 2^m < 2n.$$

Le multipôle Σ'_n cherché peut être réalisé comme sur la figure 144. On prend les $[\log_2 n] + 1$ premières sorties du multipôle Σ'_{2^m} :

$$L(\Sigma'_n) = L(\Sigma'_{2^m}) + 2 \leqslant 18\cdot 2^m + 2 \leqslant 18(2n-1) + 2 \leqslant 36n.$$

C.q.f.d.

Théorème 7. *Il existe une constante* C_0 *telle que toute formule symétrique* $S_{i_1, \ldots, i_r}(x_1, \ldots, x_n)$ *est réalisable par un circuit* $\Sigma_{S(x_1, \ldots, x_n)}$ *tel que* $L(\Sigma_{S(x_1, \ldots, x_n)}) \leqslant C_0 n$.

Démonstration. Le circuit $\Sigma_{S(x_1, \ldots, x_n)}$ cherché peut être construit comme l'indique la figure 145. Le circuit Σ'_n convertit la combinaison $(\alpha_1, \ldots, \alpha_n)$ en le code binaire $i_{(\alpha_1, \ldots, \alpha_n)}$. Le circuit Σ''_n réalise la fonction booléenne de $[\log_2 n]+1$ variables et cette

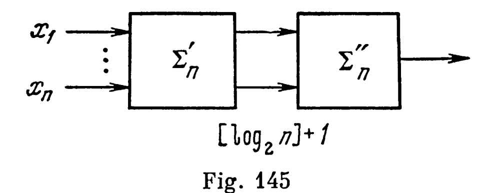


Fig. 145

fonction est égale à 1 sur les « combinaisons » $i_{(\alpha_1, \ldots, \alpha_n)}$ confondues avec $i_1, \ldots, i_r$ et à 0 dans les autres cas.

On a (d'après le théorème 5)

$$L(\Sigma_{S(x_1, \ldots, x_n)}) \leqslant 36n + 8\frac{2^{[\log_2 n]+1}}{[\log_2 n]+1} \leqslant 36n + \frac{16n}{\log_2 n} = C_0 n,$$

où l'on peut prendre $C_0 = 52$. C.q.f.d.

BIBLIOGRAPHIE

1. HUFFMAN D. A. A methods for construction of minimum-redundancy codes. — Proc. IRE, 40, N° 9, 1952, p. 1098-1101.
2. JOURAVLIEV Y. Sur la séparabilité des sous-ensembles de sommets du cube unité à n dimensions. Troudy MIAN SSSR, 51, M.; 1958, p. 143-157.
3. KOUDRIAVTSEV V. Théorème de complétude pour une classe d'automates sans contre-réactions. — Problémy kibernétiki, vyp. 8, M.; Fizmatguiz, 1962, p. 91-116.
4. KOUDRIAVTSEV V. Sur les puissances des ensembles d'ensembles précomplets de certains systèmes fonctionnels liés aux automates. Problémy kibernétiki, vyp 13, M.; Naouka, 1965, p. 45-74.
5. KOUZNETSOV A. Sur les problèmes de l'identité de la complétude fonctionnelle des systèmes algébriques. — Troudy trétégo vsésoyouznogo matématitcheskogo siezda. II. — M.; AN SSSR, 1956, p. 145-146.
6. KRATKO M. Insolubilité d'un problème de théorie des automates finis à l'aide d'un algorithme. — Diskretni analiz, vyp. 2, Novossibirsk, 1964.
7. KRATKO M. Sur l'existence de bases non récursives d'automates finis. — Alguébra i loguika, Novossibirsk: 1964, 3, N° 2, p. 33-34.
8. LIAPOUNOV A. Sur les circuits logiques de programmes. — Problémy kibernétiki, vyp. 1, M.; Fizmatguiz, 1958, p. 46-74.
9. LOUPANOV O. Sur les possibilités de synthèse de circuits d'éléments arbitraires. — Troudy MIAN SSSR, 51, 1958, p. 158-183.
10. LOUPANOV O. Sur la synthèse de certaines classes de systèmes commandés. — Problémy kibernétiki, vyp. 10, M.; Fizmatguiz, 1963, p. 88-96.
11. LOUPANOV O. Sur une approche de la synthèse des systèmes commandés ou principe de codage local. — Problémy kibernétiki, vyp. 14, M.; Naouka, p. 31-110.
12. MARKOV Al. Sur un codage alphabétique. — DAN SSSR, 1961, 132, n° 3, p. 521-523.
13. MARKOV Al. Sur un codage alphabétique. — DAN SSSR, 1961, 139, n° 3, p. 560-561.
14. MARKOV Al. Codage non récurrent. — Problémy kibernétiki, vyp. 8, M.; Fizmatguiz, 1961, p. 169-188.
15. MARTIN N. The Sheffer functions of 3-valued Logic. — J. Symb. Logic, 19, n° 1, 1954, p. 45-51.
16. McMILLAN B. Two inequalities implied by unique decipherability. — IRE Trans., IT-2, n° 4, 1956, p. 115-116.
17. ORLOV V. Une démonstration simple de l'insolubilité par un algorithme de certains problèmes de complétude des bases d'automates. — Kibernétika, 1973, n° 4, p. 109-113.
18. PICCARD S. Sur les fonctions définies dans des ensembles finis quelconques. Fund. math., 24, 1935, p. 183-185.

19. PONTRIAGUINE L. Eléments de topologie combinatoire. — M. — L.: Gostekhizdat, 1947.
20. POST E. Introduction to a general theory of elementary propositions. — Amer. J. Math., 43, 1921, p. 163-185.
21. POST E. Two-valued iterative systems of matematical logic. — Annals of Math. Studies, v. 5, Princeton Univ. Press; Princeton-London, 1941.
22. QUINE W. V. On cores and prime implicants of truth functions. — Amer. Math. Monthly, 66, n° 9, 1959.
23. ROSSER S. B., TURQUETTE A. R. Many-valued logic. — Amsterdam, 1952.
24. SALOMAA A. Some completeness criteria for sets of functons over a finite domain. I, II — Turun Ylopiston Jalkaisuja Annales Universitatis Turkuensis, sarja A53, 1962, p. 1-9; 63, 1963, p. 1-19.
25. SŁUPECKI J. Kriterium pełnosci wielowar-tosciowych systemow logiki zdan. — Comptes rendus des séances de la Société des sciences et des lettres de Varsovie, Cl. III, 32, 1939, p. 102-109.
26. TCHÊGUIS I., YABLONSKI S. Procédés logiques de contrôle du fonctionnement des circuits électriques. — Troudy MIAN SSSR, 51, 1958, p. 270-360.
27. TRACHTENBROT B. Sur la théorie des circuits à contact non itérés. — Troudy MIAN SSSR, 51, 1958, p. 226-269.
28. YABLONSKI S. Sur les superpositions des fonctions booléennes. — Matem. zb., 1952, **30** (72), p. 329-348.
29. YABLONSKI S. Constructions fonctionnelles dans la logique k-valente. — Troudy MIAN SSSR, 51, 1958, p. 5-142.
30. YABLONSKI S. Cours polycopié d'éléments de mathématiques discrètes — M.: Univ. de Moscou, 1971.
31. YABLONSKI S., GAVRILOV G., KOUDRIAVTSEV V. Fonctions booléennes et classes de Post. — M.: Naouka, 1966.
32. YANOV Y., MOUTCHNIK A. Sur l'existence de classes fermées k-valentes ne possédant pas de base finie. — DAN SSSR, 1959, **127**, n° 1, p. 44-46.

INDEX ALPHABÉTIQUE